Model Order Reduction for Parameterized Nonlinear Evolution Equations

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

von **M. Sc. Yongjin Zhang**

geb. am 06.03.1978, in Henan, China

genehmigt durch die Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg

Gutachter: **Dr. Lihong Feng**

Prof. Dr. Martin Grepl

eingereicht am: 23.03.2016
Verteidigung am: 29.06.2016

Declaration of Honor

I hereby declare that I produced this thesis without prohibited assistance and that all sources of information that were used in producing this thesis, including my own publications, have been clearly marked and referenced.

In particular I have not wilfully:

- Fabricated data or ignored or removed undesired results.
- Misused statistical methods with the aim of drawing other conclusions than those warranted by the available data.
- Plagiarised data or publications or presented them in a disorted way.

I know that violations of copyright may lead to injunction and damage claims from the author or prosecution by the law enforcement authorities.

This work has not previously been submitted as a doctoral thesis in the same or a similar form in Germany or in any other country. It hast not previously been published as a whole.

(Location, Date)

(Signature)

Acknowledgements

I would like to thank all those who supported me throughout my doctoral studies.

First and foremost, I would like to thank my supervisor Prof. Peter Benner for taking me as a Ph.D. student to work at Max Planck Institute for Dynamics of Complex Technical Systems in Magdeburg, Germany. Over the past years, I am deeply indebted to him for his strong and kind support, invaluable advice, and constructive feedback regarding publications and the thesis. I am also very grateful to my advisor Dr. Lihong Feng for her constant guidance, great patience, and numerous invaluable suggestions for my research. She helped me a lot in presentations, publications, and the thesis as well. Her kind help in my daily life deserves for a special thank, especially when I started in Germany.

I also thank Prof. Martin Grepl from RWTH Aachen University for agreeing to review the thesis. I am indebted to his valuable suggestions and comments.

I would like to thank all my colleagues that I worked with in the research group of Computational Methods in Systems and Control Theory. In particular, I am obliged to Dr. Suzhou Li and Dr. Yao Yue for fruitful discussions on some pieces of the research. Special thanks should go to Martin Hess not only for many useful discussions on our research but also for his valuable help such as translation of many documents in German, which makes my daily life easy. I am also grateful to my friends and colleagues Dr. Mian Ilyas Ahmad, Jessica Bosch, Dr. Xin Du, Dr. Ee Han, Dr. Xin Liang, Dr. Akwum Onwunta, Dr. Martin Stoll, Petar Mlinarić, Martin Redmann, and Wei Zhao for their sharing and making me not feel alone inside and outside the work. Thank you all for your kindness.

Last but not least, a very special thank goes to my family, my parents, my wife and my daughter for their never-ending support, love, and encouragement.

Large parts of this thesis have been published or are submitted for publication.

Parts of Chapter 2, e.g., the process descriptions and the numerical discretizaton of the models, were originally presented in [187, 189].

The last three sections in Chapter 3 are a revised and extended version of some contents in [188, 189].

Most parts of Chapter 4 are from some parts of [187, 188]. Section 4.2 is a revised version of [187]. The last two sections were originally presented in [188].

The first part of Chapter 5 was originally presented in [25, 187].

Most parts of Chapter 6 were originally presented [187, 188, 189].

[25]. P. Benner, L. Feng, S. Li, and Y. Zhang, *Reduced-order modeling and ROM-based optimization of batch chromatography*, in Numerical Mathematics and Advanced Applications-ENUMATH 2013, A. Abdulle, S. Deparis, D. Kressner, F. Nobile, and M. Picasso, eds., vol. 103 of Lect. Notes Comput. Sci. Eng., Springer, New York, 2015, pp. 427–435.

[187]. Y. Zhang, L. Feng, S. Li, and P. Benner, *Accelerating PDE constrained optimization by the reduced basis method: application to batch chromatography*, Internat. J. Numer. Methods Engrg., 104 (2015), pp. 983–1007.

[188]. Y. Zhang, L. Feng, S. Li, and P. Benner, *An efficient output error estimation for model order reduction of parametrized evolution equations*, SIAM J. Sci. Comput., 37 (2015), pp. B910–B936.

[189] Y. Zhang, L. Feng, A. Seidel-Morgenstern, and P. Benner, *Accelerating optimization and uncertainty quantification of nonlinear SMB chromatography using reduced-order models*. Submitted.

Abstract

Model order reduction (MOR) has emerged as an important tool in reducing the computational burden of large-scale systems, particularly in real-time or many-query contexts, e.g., optimization, control, and uncertainty quantification. In this thesis, we study projection-based MOR methods for parameterized nonlinear evolution equations. The nonlinearity and parametrization pose many difficulties in MOR, which call for efficient and sharp error bounds and effective parameter sampling strategies for constructing simulation-efficient parametric reduced-order models (ROMs) at low cost.

We propose two output error bounds to estimate the output error of the ROM in the vector space for parameterized evolution equations. One is a primal-only output error bound, and the other is a primal-dual output error bound. The former is based on the analysis of the residual of the original system, while the latter is derived by introducing and using a novel dual system. In particular, the primal-dual output error bound successfully avoids the accumulation of the residual over time, which is a common drawback in the existing error estimations for time-stepping schemes. Both error bounds are independent of the MOR methods and the spatial discretization approach employed, and they are applicable to a broad class of nonlinear and linear evolution equations.

In addition to the error bound, we pursue efficient construction of ROMs. A technique that we call adaptive snapshot selection is proposed to collect the snapshots adaptively so that the ROM can be constructed more efficiently. This technique is applicable to snapshot-based MOR methods, e.g., the reduced basis method and the proper orthogonal decomposition method. For multi-stage systems, we propose to accelerate full-order model simulation by using (intermediate) ROMs as predictors during the basis construction process.

Numerical experiments are carried out to show the performance of the proposed output error bounds and the acceleration techniques. Two academic examples are employed to test the derived error bounds, and applications to real-life models in chemical engineering are also explored. ROM-based optimization and/or uncertainty quantification of batch and simulated moving bed chromatography are successfully implemented, and the results show that the ROMs are very efficient in reducing the computational cost.

Zusammenfassung

Modellordnungsreduktion (MOR) hat sich zu einem wichtigen Werkzeug bei der Reduktion des numerischen Aufwands großskaliger Systeme entwickelt, insbesondere im Echtzeit-und Parameter-Kontext, wie z.B. Optimierung, Steuerung und Quantifizierung von Unsicherheiten. In dieser Arbeit werden projektionsbasierte MOR Methoden für parametrisierte, nichtlineare Evolutionsgleichungen untersucht. Die Nichtlinearität und Parametrisierung stellt zahlreiche Anforderungen an die MOR, wie effiziente und genaue Fehlerschranken und effektive Parameter-Sampling Strategien um simulatonseffiziente parametrische reduzierte Modelle mit geringem numerischen Aufwand zu konstruieren.

Hier werden zwei Fehlerschranken im Ausgang vorgestellt, die den Ausgangfehler des reduzierten Modells im Vektorraum der parametrisierten Evolutionsgleichungen schätzen. Zum Einen eine primale Ausgangfehlerschranke und zum Anderen eine primal-duale Ausgangfehlerschranke. Erstere basiert auf dem Residuum des originalen Systems, während Letztere mit Hilfe eines neuartigen dualen Systems hergeleitet wird. Insbesondere die primal-duale Fehlerschranke verhindert erfolgreich die Akkumulation des Residuums über die Zeit, was bei den existierenden Fehlerschätzern für zeitabhängige Probleme ein häufiges Problem ist. Beide Fehlerschranken sind unabhängig von der MOR Methode und dem räumlichen Diskretisierungsansatz und lassen sich auf eine große Klasse von nichtlinearen und linearen Evolutionsgleichungen anwenden.

Zusätzlich zur Fehlerschranke wird die effiziente Konstruktion reduzierter Modelle untersucht. Eine adaptive Snapshot-Auswahl wird vorgestellt um Snapshots adaptiv auszuwählen, sodass das reduzierte Modell effizienter konstruiert werden kann. Diese Technik lässt sich auf Snapshot-basierte MOR Methoden anwenden, wie z.B. die Reduzierte Basis Methode und die Proper Orthogonal Decomposition Methode. Mit mehrstufigen Systemen wird die großskalige Modellsimulation durch reduzierte (dazwischenliegende) Modelle beschleunigt, indem die reduzierten Modelle während der Basiskonstruktion als Prädiktor dienen.

Numerische Experimente werden durchgeführt, um den Vorteil der Ausgangfehlerschranken und der Beschleunigungstechniken zu zeigen. Zwei akademische Beispiele nutzen die hergeleiteten Fehlerschranken und Anwendungen im in der chemischen Verfahrenstechnik werden ebenfalls untersucht. Optimierung und Quantifizierung von Unsicherheiten

basierend auf reduzierten Modellen in der ‘Batch’ und ‘Moving Bed’ Chromatographie ist implementiert und zeigt, dass die reduzierten Modelle den numerischen Aufwand sehr effizient reduzieren.

List of Acronyms

ASS	adaptive snapshot selection
BPIM	best point interpolation method
BT	balanced truncation
CRB	collateral reduced basis
CSS	cyclic steady state
DEIM	discrete empirical interpolation method
DOF	degree of freedom
EI	empirical interpolation
EIM	empirical interpolation method
FOM	full-order model
HSV	Hankel singular value
LDEIM	localized discrete empirical interpolation method
MEMS	micro-electro-mechanical systems
MOR	model order reduction
RB	reduced basis
RBM	reduced basis method
ROM	reduced-order model
PDE	partial differential equation
PMOR	parametric model order reduction
(P)MOR	(parametric) model order reduction

POD	proper orthogonal decomposition
SMB	simulated moving bed
SpF	speedup factor
SVD	singular value decomposition
UQ	uncertainty quantification
w.r.t.	with respect to

List of Symbols

Symbols for Set, Space, Matrix, and Operators

$\mathbb{N}$	set of positive integers
$\mathbb{R}$	set of real numbers.
$\mathbb{C}$	set of complex numbers
$\mathbb{R}^{m\times n}$	vector space of real matrices with m rows and n columns
$\mathbb{C}^{m\times n}$	vector space of complex matrices with m rows and n columns
I_m, I	identity matrix of $m \times m$ or suitable size
e_i	ith column of the identity matrix I
A	matrix with suitable size
A^T	transpose of matrix A
$\mathrm{diag}(a_1, \ldots, a_m)$	diagonal matrix with diagonal entries $a_1 \ldots, a_m$
$\sigma_{\max}(A)$	maximal singular value of a matrix A
$\lambda_{\max}(A)$	maximal eigenvalue of a matrix A
$\langle \cdot, \cdot \rangle$	inner product
μ	parameter or parameter vector
Ω	computational domain of spatial variables
$\mathcal{P}$	domain of parameters

Symbols for Numerical Discretization

Δt	time step size
Δx	spatial grid size

K	total number of time steps
t^n	time instance at $t = t^n$, $n = 0, 1, \ldots, K$
$\mathcal{N}$	dimension or size of the full-order model
N	dimension or size of the reduced-order model
M	dimension of the interpolation basis
n_{I}	number of inputs
n_{O}	number of outputs

Symbols for Chromatography Models

L	length of a chromatographic column
A_c	cross-section area of a chromatographic column
c_z	concentration of component z in the liquid phase, $z = a, b$
q_z	concentration of component z in the solid phase, $z = a, b$
κ_z	mass-transfer coefficient for component z
q_z^{Eq}	adsorption equilibrium
Q	volumetric flow rate
t_{s}	time of switching period
t_{in}	time of injection period
$P_{\mathrm{s}}[\cdot]$	column-wise switching operator
ϵ	column porosity
Pe	Péclet number
Pr	production rate
Pu	product purity
Rec	recovery yield

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation

Numerical simulation has been playing an important role in computational science and engineering. Results from numerical simulation of a physical model provide a better understanding of physical phenomena and guide engineers to design more reliable experiments, which significantly shortens the design cycle and reduces the cost. As the complexity of physical models considered increases, mathematical modeling and numerical simulation of those models become more and more challenging. The current computer architecture and algorithms cannot satisfy the increasing demand especially in many time-critical applications, although great improvements have been made in both aspects during the past decades. These time-critical applications that are relevant to this thesis include the following two scenarios: the real-time context and the many-query context. The former refers to problems that require the simulation to be done very fast, e.g., real-time control, real-time prediction, and other simulation-based decision-making processes, while the latter refers to the case that the simulation needs to be repeatedly performed under certain differ-

ent (input) conditions, e.g., optimal design, optimization, and uncertainty quantification (UQ).

In many applications, the physical model is described by partial differential equations (PDEs), which are often parameterized. The parameters may be introduced in many ways, representing material properties, geometry configurations, initial conditions, boundary conditions, source or force terms, etc. Typically, the quantity of interest is not the field variable (the solution to the system) but rather a certain output that is a function of the field variable. In fact, the output of interest is usually a physical quantity that is used to measure or assess the behavior of the system corresponding to different parameters, e.g., lift/drag, maximal temperature, and purity of products.

In most cases, the analytical solution to the PDEs is not available, and numerical solution is considered instead. Thus, the PDEs must be discretized with a suitable numerical method, e.g., the finite element method, the finite volume method, and the finite difference method [151]. Taking a time-dependent PDE as an example, two approaches are often adopted: the semi-discretized approach and the fully discretized approach. The former approach first discretizes the PDE in space to yield a semi-discrete system

$$F(\dot{u}(t,\mu), u(t,\mu), t, \mu, p(t)) = 0, \tag{1.1}$$

where $u(t,\mu) \in \mathbb{R}^{\mathcal{N}}$ is the field variable or the state, $\dot{u}(t,\mu) = \frac{\mathrm{d}u(t,\mu)}{\mathrm{d}t}$ is the time derivative, $p(t) \in \mathbb{R}^{n_{\mathrm{I}}}$ is a time-dependent input. Then, a certain time-integration scheme is employed to acquire the solution at the time of interest. The latter approach discretizes the PDE in time and space simultaneously to yield a fully discrete system

$$F(u^{n+1}(\mu), u^n(\mu), \mu, p(t^n)) = 0, \tag{1.2}$$

and the solution at the time of interest is obtained by solving (1.2) step by step. Here, $u^n(\mu) \in \mathbb{R}^{\mathcal{N}}$ is the numerical solution at the time $t = t^n$. Once the field variable $u(t,\mu)$ or $u^n(\mu)$ is computed, the output of interest, $y(t,\mu) = \ell(u(t,\mu))$ or $y^n(\mu) = \ell(u^n(\mu))$, is easily obtained.

In numerical simulations, to capture the dynamics of the system precisely, the corresponding discretized system is of a large order (size)—the number of degrees of freedom (DOFs), e.g., $\mathcal{N}$ is in $\mathcal{O}(10^6)$ for simulation of some three-dimensional problems. Such a large-scale system needs to be simulated many times in the aforementioned many-query contexts. A

single simulation of such a system may take minutes, hours, or even days, let alone many repeated simulations, which can be prohibitively time-consuming. In real-time analysis, the output response needs to be obtained in a limited amount of time, say, a few seconds or even less. These time-critical applications call for system approximations by a surrogate model that is not only of reduced scale and complexity but also able to capture the main features of the original large-scale complex model.

Model order reduction (MOR), also known as model reduction or dimension reduction, is a powerful technique for constructing a low-cost approximation of a large-scale system that results from the discretization of PDEs. On the one hand, the low-cost approximation should have the same structure as the original large-scale system but be with a much smaller order; on the other hand, it must have acceptable accuracy for the input-output representation of the original system. Conventionally, we call the low-cost approximation the reduced-order model (ROM), and accordingly, the original large-scale system is called the full-order model (FOM). For example, a linear case of (1.1) that is often employed in MOR appears as a parametric dynamical system

$$E(\mu)\dot{u}(t,\mu) = A(\mu)u(t,\mu) + B(\mu)p(t), \tag{1.3}$$

where $E(\mu), A(\mu) \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, and $B(\mu) \in \mathbb{R}^{\mathcal{N}\times n_{\mathrm{I}}}$ are the coefficient matrices. The ROM is formulated as

$$\hat{E}(\mu)\dot{u}_r(t,\mu) = \hat{A}(\mu)u_r(t,\mu) + \hat{B}(\mu)p(t), \tag{1.4}$$

where $u_r(t,\mu) \in \mathbb{R}^N$ is the vector of unknowns of the ROM, and $\hat{E}(\mu), \hat{A}(\mu) \in \mathbb{R}^{N\times N}$, $\hat{B}(\mu) \in \mathbb{R}^{N\times n_{\mathrm{I}}}$ are the reduced matrices, and $N \ll \mathcal{N}$. Due to its small size and negligible error, the derived ROM is used as a surrogate model of the large-scale system for different purposes, e.g., design, optimization, control, and UQ. Applications of ROMs can be found in various disciplines, such as micro-electro-mechanical systems (MEMS), fluid and solid mechanics, structural mechanics, acoustics, circuit design, image processing, etc.

During the past decades, various MOR methods have been developed in different disciplines. Moreover, many of these are described in different terms due to some specific applications even though they share many common features and origins in principle. Roughly speaking, those MOR methods include balanced truncation (BT) [19, 77, 125], Krylov subspace methods (also known as moment-matching methods [65], or Padé approximation methods [60]) [14, 24, 73], proper orthogonal decomposition (POD) [35, 108, 153, 171, 179, 180, 181], and reduced basis methods (RBMs) [81, 100, 133, 140, 146, 162].

Based on whether the parameters can be reliably kept as symbolical quantities in the resulting ROMs, these MOR methods can be simply classified into non-parametric MOR methods and parametric MOR (PMOR) methods. Non-parametric MOR methods for linear time-invariant systems have been intensively studied in the past years; see, e.g., [9, 19] and references therein. PMOR, however, has emerged recently as an important research area, see a comprehensive survey in [26]. Recent development of various (parametric) MOR ((P)MOR) methods and their applications can be found in [9, 19, 22, 26, 27, 140, 149, 167] and references therein.

As mentioned earlier, the development of (P)MOR methods is often driven by applications. Here, we are motivated by batch chromatography and simulated moving bed (SMB) chromatography, which are two major separation processes in chemical engineering and will be introduced in Chapter 2. These two processes are described by time-dependent nonlinear convection-diffusion equations, and both require a long-time integration process. In addition, the SMB process is a multi-stage system with periodic switching. Optimization of chromatography is of great importance since it allows to exploit the full economic potential of the process and reduce the separation cost, reflected by many studies on this area in the literature [11, 12, 50, 51, 67, 74, 78, 104, 112, 113, 152, 168, 173, 174]. Note that most of them are based on the FOM (i.e., the high-fidelity model), which are expensive and call for efficient approximation strategies.

In this thesis, we pursue PMOR methods for parameterized nonlinear evolution equations, exemplified by the aforementioned chromatographic processes. PMOR is designed for a broad class of problems in which the governing equations depend on a set of parameters. Using PMOR methods, the parameters in the FOM are kept as symbolic quantities in the ROM so that the resulting parametric ROM is globally reliable, i.e., it is able to reproduce the main dynamics of the FOM corresponding to any variation of the parameter in the whole parameter domain. To construct ROMs for parametric systems, two correlated questions come up immediately. One is how to measure or estimate the error between the reduced approximation and the reference quantity computed from the FOM, namely, error control or error bound/estimation. The other is how to collect information efficiently to build the ROM, which is essentially related to effective parameter sampling.

Although significant progress has been made in both aspects (error control and parameter sampling) during the past decades, many problems still exist. For example, in applications

to multi-stage systems like the aforementioned SMB chromatography, existing PMOR methods cannot work well without proper error estimation because the evolution process is extremely complicated. Actually, the existing *a posteriori* error bounds or estimations often fail, especially when periodic switching is involved and/or a very large number of time steps are employed. In addition, constructing (globally) accurate ROMs for these problems is very expensive, which limits the reduction efficiency concerning the total computational cost. This thesis dedicates itself to derive efficient *a posteriori* error bounds and develop effective sampling strategies for collecting snapshots.

1.2 Strategy and objectives

Generally, PMOR methods permit an upfront process to construct a ROM in the offline stage and implement the simulation based on the ROM in the online stage, namely, the offline-online decomposition. More precisely, all high-dimension dependent quantities are precomputed and stored during the offline stage. This process could be expensive but needs to be performed only once. During the online stage, the computation of the output response only depends on the ROM for any given feasible parameter value. In other words, it is independent of the (high) dimension of the FOM. Therefore, the output response can be obtained rapidly. We aim to seek efficient strategies for constructing ROMs that can be simulated fast online, and meanwhile the offline computation is at low cost.

Note that the offline-online decomposition can be easily implemented for problems that are linear or have affine expressions, i.e., they have parameter-separable forms. However, for systems with nonlinearity (w.r.t. the field variable) and/or nonaffine parameter dependence, the cost of evaluating the nonlinear and/or nonaffine parts cannot be reduced by projection. To tackle the problem, the empirical interpolation method [17] or its variants (e.g., the discrete empirical interpolation method [45] or empirical operator interpolation [49]) can be employed. With the additional techniques, the complexity of those parts can be reduced by interpolation, and thus the offline-online computation can be implemented efficiently.

For parametric systems, the reduced basis (RB), used to construct the ROM, is often built iteratively via a greedy algorithm, which will be further addressed in detail in Section 3.3. The greedy algorithm adaptively selects a parameter that causes the largest error (mea-

sured by the true error or some error estimations) in the parameter domain or its surrogate training set, to update the RB until the resulting ROM meets the accuracy requirement.

Since computing the true error involves the simulation of the FOM, which is often very expensive, an *a posteriori* error estimation is considered instead. Actually, for linear or time-independent problems, the error bound has been intensively investigated [72, 85, 147, 158, 159, 39, 177, 176]. However, the output error bound for nonlinear time-dependent problems is less explored; see, e.g., [49, 132]. In this thesis, we pursue efficient *a posteriori* output error estimations for parameterized nonlinear evolution equations. Based on the analysis of the residual of the ROM, we derive an output error bound for nonlinear evolution equations in the vector space. The derived error bound is efficient and applicable to any (spatial) discretization approach employed for the system. However, this error bound is based only on the primal system (the original system) and may lose sharpness for some problems. This implies that the size (order) of the ROM might be unnecessarily enlarged when such *a posteriori* bound is employed in the greedy algorithm during the basis construction process. Thus, we seek to derive a more accurate error bound by a novel dual approach. The derived output error bound is proved to be much sharper. Both output error bounds are addressed in detail in Chapter 4.

We now come to the strategies for accelerating the offline computations. Collecting snapshots is important for snapshot-based MOR methods to efficiently construct a ROM that is qualified over a wide range of parameters. The snapshots used to compute the basis are often selected from the solutions at different samples in parameters and/or in different time instances (for time-dependent problems). Many efforts have been devoted in devising effective methods for parameter sampling [53, 82, 83, 34, 141]. However, little attention has been paid to the selection of the solutions from different time instances. Actually, the total number of time steps might be large, e.g., in the simulation of batch chromatography. In such a case, if the solutions at all time steps are taken as snapshots, the subsequent computation (e.g., singular value decomposition) will be expensive since the number of snapshots is too large; if we just trivially select parts of the solutions, e.g., at every two or several time steps, the resulting RB approximation might be of low accuracy because important information may have been lost due to such a naive snapshot selection. In this thesis, we introduce a technique that we call adaptive snapshot selection (ASS) to collect snapshots in an efficient way. That is, we adaptively collect snapshots by discarding the redundant ("almost" linearly dependent) information according to the variation of the time

trajectory. With ASS, we can obtain a more compact snapshot set containing only the most representative information of the solution. Consequently, the runtime for computing the ROM and the memory requirement can be significantly reduced.

Another strategy for accelerating the offline computation is devised for multi-stage systems. Note that an SMB process is a multi-stage system driven by a periodic switching procedure. Due to the periodic switching, the system has a cyclic steady state rather than a steady state. We carefully study the properties of the multi-stage system and propose to accelerate FOM simulation using ROMs as predictors during the construction of the RB. We call the technique ROM prediction since the ROM is used to predict a "good" initial state so that the time of the FOM simulation is reduced at each iteration of the RB extension process. The two accelerating techniques, ROM prediction and ASS, will be presented in detail in Chapter 5.

1.3 Accomplishments

(P)MOR can be formulated in different ways, e.g., in a semi-discrete form, in a fully discretized form, or in a weak form of PDEs in the functional space. We opt for the fully discretized framework defined in (1.2). All the analyses and derivations are done in a finite-dimensional vector space. The theories and the techniques presented in this thesis are independent of the spatial discretization employed for the FOM. The main accomplishments of this thesis are summarized as follows:

1. Output error bounds. Two output error bounds are derived for projection-based MOR methods in the vector space for parameterized (nonlinear) evolution equations. One is a primal-only error bound, and the other is a primal-dual error bound. The former is more efficient, while the latter is sharper. They are independent of discretization approach employed and can be used to guide the parameter sampling for the construction of the RB. The derived error bounds are tested by two academic examples and two real-life models from chemical engineering, which indicate that they are applicable to a broad class of parameterized evolution equations.

2. Adaptive snapshot selection (ASS). The ASS technique is proposed to reduce the offline time for constructing the basis for snapshot-based MOR methods, e.g., the

POD method and the RBM.

3. ROM prediction. Accelerating FOM simulation by using ROMs as predictors during the RB extension process. This can be applied to multi-stage systems so that the final cyclic steady state of the SMB chromatography, can be obtained more quickly.

4. ROM-based optimization. The resulting ROMs are employed to efficiently implement optimization of chromatography in chemical engineering. The strength of the PMOR method is demonstrated by the industry-relevant applications.

5. ROM-based UQ. It is the first time that the parametric ROMs are employed to analyze the robustness of a nonlinear SMB process with flow rate uncertainty.

1.4 Outline

This thesis is organized as follows. Chapter 2 addresses the motivating examples in chemical engineering, namely, batch chromatography and SMB chromatography. The FOMs of both processes are constructed by the finite volume method. The optimization of batch and SMB chromatography is presented, and the methods of surrogate-based optimization are discussed.

The methodology of (P)MOR is presented in Chapter 3. Various (P)MOR methods are reviewed, and differences and similarities are also discussed. In addition, we introduce the RBM for parameterized nonlinear time-dependent problems. To deal with the nonlinearity or non-affinity, the empirical interpolation method and related techniques are addressed. For the RB construction, we address the greedy algorithm and the POD-Greedy algorithm. The offline-online decomposition strategy is discussed.

Chapter 4 presents *a posteriori* output error bound/estimation for parameterized nonlinear evolution equations. One is based on the analysis of the residual of the primal system, and the other adopts a primal-dual approach, where a novel dual system is introduced to aid the derivation of the output error estimation. Two academic examples are employed to show the performance of the derived error bound/estimation.

In Chapter 5, we address the techniques for accelerating the offline computation: ASS and ROM prediction. These two techniques can be easily implemented to reduce the offline

computational cost.

Numerical experiments for batch and SMB chromatography are carried out in Chapter 6. The performance of the proposed error bounds and the accelerating techniques is demonstrated. The application of the generated parametric ROM in optimization and UQ are explored.

Chapter 7 concludes this thesis. Some perspectives for future work are given in the end.

Chapter 2

Efficient Optimization of Chromatography via Reduced-order Modeling

Contents

In this chapter, we introduce two motivating examples: discontinuous batch chromatography and continuous SMB chromatography, which are widely used for separation and purification in chemical engineering. Optimal design, real-time control, and optimization of batch and SMB chromatography are of great importance in chemical engineering. We show the mathematical modeling of both processes and the corresponding optimization problems. Parts of the contents in this chapter were originally presented in [187, 189].

2.1 Model description of chromatography

In this section, we introduce two major separation processes in chemical engineering: batch chromatography and SMB chromatography. We show the basic physical process and the mathematical modeling of the two chromatographic processes. These real-life models are used to test the methods and techniques presented in this thesis and will be revisited in Chapters 5 and 6.

2.1.1 Batch chromatography

Batch chromatography, as a crucial separation and purification tool, is widely employed in food, fine chemical, and pharmaceutical industries. The principle of batch elution chromatography for binary separation is shown schematically in Figure 2.1. During the injection period t_{in}, a mixture consisting of a and b is injected from the inlet of the column which is packed with a suitable stationary phase. With the help of the mobile phase, the feed mixture flows through the column. Since the solutes to be separated exhibit different adsorption affinities to the stationary phase, they move at different velocities in the column and thus separate from each other when exiting the column. At the column outlet, component a is collected between cutting points t_3 and t_4, and component b is collected between t_1 and t_2. Here the positions of t_1 and t_4 are determined by a minimum concentration threshold that the detector can resolve, and the positions of t_2 and t_3 are determined by the purity specifications (Pu_a and Pu_b) imposed on the products. After a cycle period $t_{\text{cyc}} := t_4 - t_1$, the injection is repeated.

The dynamic behavior of the chromatographic process is described by an axially dispersed plug-flow model with limited mass-transfer rate characterized by a linear driving force approximation. The governing equations in the dimensionless form are formulated as follows:

$$\begin{cases} \dfrac{\partial c_z}{\partial t} + \dfrac{1-\epsilon}{\epsilon}\dfrac{\partial q_z}{\partial t} = -\dfrac{\partial c_z}{\partial x} + \dfrac{1}{Pe}\dfrac{\partial^2 c_z}{\partial x^2}, & 0 < x < 1, \\ \dfrac{\partial q_z}{\partial t} = \dfrac{L}{Q/(\epsilon A_c)}\kappa_z \left(q_z^{\text{Eq}} - q_z\right), & 0 \le x \le 1, \end{cases} \tag{2.1}$$

where c_z, q_z are the concentrations of the component $z\,(z = a, b)$ in the liquid and solid phase, respectively, Q the volumetric feed flow rate, A_c the cross-sectional area of the

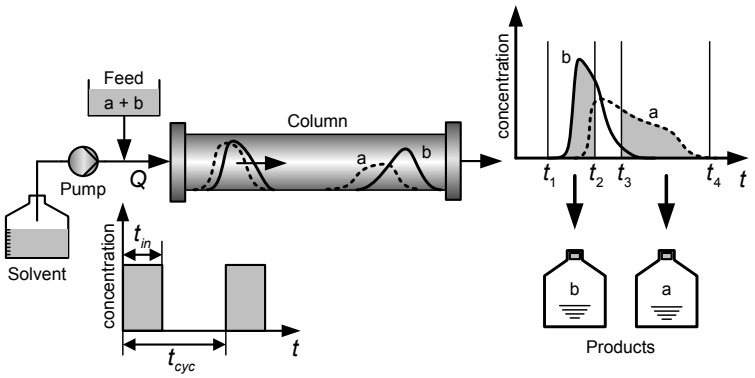


Figure 2.1: Sketch of a batch chromatographic process for the separation of a mixture of two components a and b.

column with the length L, ϵ the column porosity, κ_z the mass-transfer coefficient, and Pe the Péclet number. The adsorption equilibrium q_z^{Eq} can be described by different types of isotherm equations. Here, it is described by the isotherm equations of bi-Langmuir type,

$$q_z^{\mathrm{Eq}} = f_z(c_a, c_b) := \frac{H_{z1}c_z}{1 + K_{a1}c_a^{\mathrm{f}}c_a + K_{b1}c_b^{\mathrm{f}}c_b} + \frac{H_{z2}c_z}{1 + K_{a2}c_a^{\mathrm{f}}c_a + K_{b2}c_b^{\mathrm{f}}c_b}, \tag{2.2}$$

where c_z^{f} is the feed concentration of component z, and H_{zj}, K_{zj} are the Henry constants and thermodynamic coefficients, respectively. The initial and boundary conditions are given as follows:

$$\begin{cases} c_z(0,x) = 0, \quad q_z(0,x) = 0, \quad 0 \le x \le 1, \\ \dfrac{\partial c_z}{\partial x}|_{x=0} = Pe\left(c_z(t,0) - \chi_{[0,t_{\mathrm{in}}]}(t)\right), \\ \dfrac{\partial c_z}{\partial x}|_{x=1} = 0, \end{cases} \tag{2.3}$$

where t_{in} is the injection period, and $\chi_{[0,t_{\mathrm{in}}]}$ is the characteristic function,

$$\chi_{[0,t_{\mathrm{in}}]}(t) = \begin{cases} 1 & \text{if } t \in [0, t_{\mathrm{in}}], \\ 0 & \text{otherwise.} \end{cases}$$

More details about the mathematical modeling for batch chromatography, e.g., the dimensional form and physical descriptions can be found in [78].

Note that the feed flow rate Q and the injection period t_{in} are often considered as the operating variables, denoted as $\mu := (Q, t_{\mathrm{in}})$, which play the role of parameters in the PDEs (2.1)−(2.3). The system of PDEs is nonlinear, time-dependent and has nonaffine

parameter dependence. The nonlinearity of the system is reflected by (2.2). To capture the system dynamics precisely, a large number of DOFs must be introduced for the discretization of the PDEs. Moreover, the interesting range of the Péclet number is large, e.g., in the order of $\mathcal{O}(10^3)$, which implies the underlying problem is convection dominated.

Batch chromatography is a useful separation process, and it has several merits as follows: 1) it is very flexible to be implemented; 2) several (more than two) components from a mixture can be separated during one separation process; 3) varying compositions of the desorbent can be used to enhance separation efficiency. However, it is typically performed discontinuously, which limits its applicability of large-scale separation processes. By contrast, SMB chromatography has gained increasing popularity because of its advantages in terms of productivity and solvent consumption. An intensive discussion of batch and SMB chromatography can be found in [172]. In the next subsection, we come to the SMB chromatography.

2.1.2 SMB chromatography

SMB technology was developed by universal oil products in early 1960s [32] and has been traditionally used in oil and energy industry for recovery and purification of p-xylene and other aromatic components separated from naphtha feed. After it became commercialized heavily in the oil industry, SMB technology made its mark on the food industry by separating fructose from glucose in a molasses feed. Nowadays, it is widely used in food, fine chemical, pharmaceutical industries. A review of SMB chromatography can be found in [168].

An SMB unit typically consists of several identical chromatographic columns connected in a series, as shown in Figure 2.2. Four ports divide the SMB unit into four zones, which play different roles in a separation process. The mixture to be separated and the eluent are fed through the two inlets, and the two purified components are withdrawn from the two outlets, respectively. These processes are performed continuously, and the separation regime is accomplished through a counter current movement of the liquid and solid phase by shifting the inlet and outlet ports one column ahead in the direction of the fluid flow in a certain switching period t_{s}. For more details, we refer to [12, 152, 168].

The main dynamics of the fluid flow in all columns in an SMB unit are the same except for

the node balance relations. We first address the mathematical modeling of the dynamics in one chromatographic column, and then describe the node balance equations in between. In this work, we assume that the dynamics of each chromatographic column is described by an axially dispersed plug flow model with a limited mass-transfer rate characterized by a linear driving force approximation. The mass balance in the column k ($k = 1, \ldots, N_{\text{col}}$) can be given by

$$\frac{\partial c_{z,k}}{\partial t} + \frac{1-\epsilon}{\epsilon}\frac{\partial q_{z,k}}{\partial t} = -\frac{Q_k t_{\text{s}}}{\epsilon A_c L}\left(\frac{\partial c_{z,k}}{\partial x} - \frac{1}{Pe}\frac{\partial^2 c_{z,k}}{\partial x^2}\right), \quad 0 < x < 1, \tag{2.4}$$

where $c_{z,k}, q_{z,k}$ are the concentrations of the component z ($z = a, b$) in the liquid and solid phase in the kth column, respectively, Q_k the flow rate, A_c the cross-sectional area of the column, L the column length, ϵ the column porosity, and Pe the Péclet number. Note that t and x are in the dimensionless form, i.e., $t = t/t_{\text{s}}$, $x = x/L$. The adsorption rate is described by the linear driving force approximation as follows:

$$\frac{\partial q_{z,k}}{\partial t} = t_{\text{s}}\kappa_z(q_{z,k}^{\text{Eq}} - q_{z,k}), \quad 0 \le x \le 1, \tag{2.5}$$

where κ_z is the mass-transfer coefficient and $q_{z,k}^{\text{Eq}}$ is the adsorption equilibrium defined by the adsorption isotherm function

$$q_{z,k}^{\text{Eq}} = f_z(c_{a,k}, c_{b,k}). \tag{2.6}$$

Different separation processes are described by different adsorption isotherm equations. Two types of isotherm equations will be studied. One is of the bi-Langmuir type defined in (2.2), and the other is described by a linear function. They will be further detailed in the numerical experiments in Chapter 6.

The Danckwerts type boundary conditions are imposed to equation (2.4), i.e.,

$$\begin{cases} \dfrac{\partial c_{z,k}}{\partial x}|_{x=0} = Pe(c_{z,k}(t,0) - c_{z,k}^{\text{in}}(t)), \\ \dfrac{\partial c_{z,k}}{\partial x}|_{x=1} = 0, \end{cases} \tag{2.7}$$

where $c_{z,k}^{\text{in}}(t)$ is the concentration of component z at the inlet of column k. The system can be completed by some initial conditions,

$$c_{z,k}(0,x) = c_{z,k}^0, \quad q_{z,k}(0,x) = q_{z,k}^0, \quad k = 1, \ldots, N_{\text{col}}.$$

By mass conservation law, the concentration at the inlet of a column is equal to the concentration at the outlet of the previous column, i.e.,

$$c^{\mathrm{in}}_{z,k+1}(t) = c_{z,k}(t,1), \tag{2.8}$$

except for the feed and desorbent nodes. The mass balances at the inlet and outlet ports can be established as follows:

Desorbent node:

$$\begin{aligned} Q_{\mathrm{I}} &= Q_{\mathrm{IV}} + Q_{\mathrm{D}}, \\ Q_{\mathrm{I}} c^{\mathrm{in}}_{z,1}(t) &= Q_{\mathrm{IV}} c_{z,N_{\mathrm{col}}}(t,1). \end{aligned} \tag{2.9}$$

Extract node:

$$\begin{aligned} Q_{\mathrm{II}} &= Q_{\mathrm{I}} - Q_{\mathrm{E}}. \\ c^{\mathrm{in}}_{z,k_{\mathrm{I}}+1}(t) &= c_{z,k_{\mathrm{I}}}(t,1). \end{aligned} \tag{2.10}$$

Feed node:

$$\begin{aligned} Q_{\mathrm{III}} &= Q_{\mathrm{II}} + Q_{\mathrm{F}}, \\ Q_{\mathrm{III}} c^{\mathrm{in}}_{z,k_{\mathrm{I}}+k_{\mathrm{II}}+1}(t) &= Q_{\mathrm{II}} c_{z,k_{\mathrm{I}}+k_{\mathrm{II}}}(t,1) + c^{\mathrm{F}}_z Q_{\mathrm{F}}. \end{aligned} \tag{2.11}$$

Raffinate node:

$$\begin{aligned} Q_{\mathrm{IV}} &= Q_{\mathrm{III}} - Q_{\mathrm{R}}, \\ c^{\mathrm{in}}_{z,k_{\mathrm{I}}+k_{\mathrm{II}}+k_{\mathrm{III}}+1}(t) &= c_{z,k_{\mathrm{I}}+k_{\mathrm{II}}+k_{\mathrm{III}}}(t,1). \end{aligned} \tag{2.12}$$

Here, k_J is the number of columns in the zone J, Q_J is the flow rate in the corresponding zone, $J \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}\}$, c^{F}_z is the feed concentration of the solute z, and Q_{D}, Q_{E}, Q_{F}, Q_{R} are the desorbent, extract, feed, and raffinate flow rate, respectively. Note that the flow rates within the columns in each zone are assumed to be uniformed. For example, for an SMB unit with eight columns and 2-2-2-2 configurations, the flow rate Q_k in the kth column satisfies the following relations:

$$Q_1 = Q_2 = Q_{\mathrm{I}}, \quad Q_3 = Q_4 = Q_{\mathrm{II}}, \quad Q_5 = Q_6 = Q_{\mathrm{III}}, \quad Q_7 = Q_8 = Q_{\mathrm{IV}}.$$

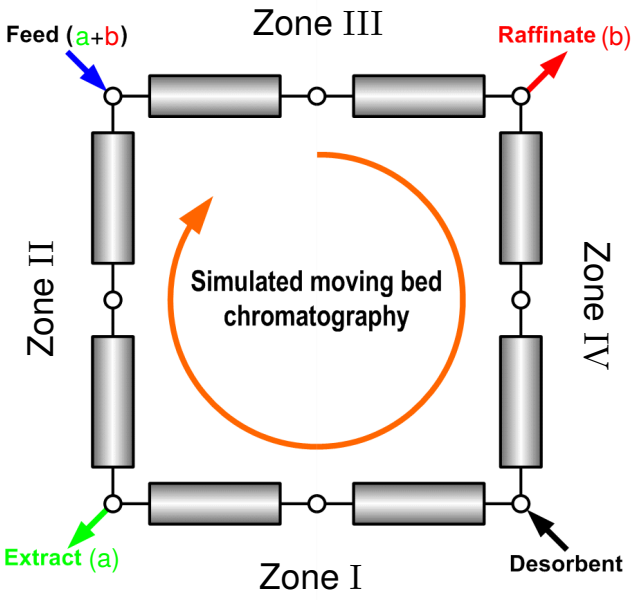


Figure 2.2: Schematic illustration of an SMB chromatographic process with four zones and eight columns.

The regime of the SMB system is a cyclic steady state (CSS), rather than a steady state, due to the periodic switching procedure. That is, during the CSS period, the concentration profiles are still varying over time, but they are identical between two consecutive switching periods. For numerical simulation, when the time-stepping scheme is employed, the system is simulated cycle by cycle, at the end of each period the state undergoes a shift and the shifted state acts as the initial state for the next period. This process continues until the CSS condition is reached. The switching procedure is expressed as

$$c_{z,\mathsf{T}+1}(0,x) = P_{\mathrm{s}}[c_{z,\mathsf{T}}(1,x)], \quad q_{z,\mathsf{T}+1}(0,x) = P_{\mathrm{s}}[q_{z,\mathsf{T}}(1,x)], \quad \mathsf{T} = 1,2,\ldots, \tag{2.13}$$

where $P_{\mathrm{s}}[\cdot]$ is a column-wise switching operator, and T refers to the Tth period. The CSS condition can be defined by, e.g.,

$$\max_{z=\{a,b\}} \max\left\{\|c_z(0,\cdot) - P_{\mathrm{s}}[c_z(1,\cdot)]\|, \|q_z(0,\cdot) - P_{\mathrm{s}}[q_z(1,\cdot)]\|\right\} < \varepsilon_{\mathrm{CSS}}, \tag{2.14}$$

where $\varepsilon_{\mathrm{CSS}}$ is a user-specified CSS tolerance. The CSS condition in (2.14) shows that the concentrations within the SMB unit at the end of a period are almost the same as those at

the beginning of the next period (the period apart from one column shift). Alternatively, check whether the output of interest, e.g., the purity of products in two consecutive periods, are equal.

From the prospect of practical implementations, the automation of SMB chromatography is far more complex compared to batch chromatography, e.g., the multi-switching procedure is involved in the SMB process, which is difficult to handle. From the prospect of numerical simulations, the two processes are described by a coupled system of convection-diffusion equations, and a proper discretization should be employed, as addressed in the following section.

2.2 Numerical discretization

As mentioned previously, both models are described by a coupled system of convection-diffusion equations. Moreover, the interesting range of the Péclet number is usually large, which means that the system is convection dominated, and the solution to the system may have stiff profiles. We use the finite volume method for the discretization, by which the conservation property of the underlying systems can be well preserved.

2.2.1 Numerical discretization of the batch chromatographic model

We use the finite volume method to discretize the batch chromatographic model (2.1)−(2.3), where the Lax–Friedrichs flux [111] is used to solve the convection flux, and the central difference scheme is applied to evaluate the diffusion flux. For the temporal discretization, we use the Crank–Nicolson scheme, which yields a second-order accurate evolution scheme. Let Δt and Δx be the properly chosen temporal step and spatial grid size, respectively. The fully discretized finite volume formulation for the system (2.1) can be written as follows:

$$\begin{cases} Ac_z^{n+1} = Bc_z^n + d_z^n - \dfrac{1-\epsilon}{\epsilon}\Delta t h_z^n, \\ q_z^{n+1} = q_z^n + \Delta t h_z^n, \quad z = a, b, \end{cases} \tag{2.15}$$

where $c_z^n := c_z^n(\mu) = ({c_z}_1^n, \ldots, {c_z}_{\mathcal{N}}^n)^T$, $q_z^n := q_z^n(\mu) = ({q_z}_1^n, \ldots, {q_z}_{\mathcal{N}}^n)^T \in \mathbb{R}^{\mathcal{N}}$ stand for the solutions of the field variables c_z and q_z at time instance $t^n := n\Delta t\,(n = 0, \ldots, K)$,

$A, B \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ are tridiagonal constant matrices, d_z^n and h_z^n are parameter- and time-dependent,

$$d_z^n := d_0^n e_1 \in \mathbb{R}^{\mathcal{N}}, \quad h_z^n := (h_{z1}^n, \ldots, h_{z\mathcal{N}}^n)^T \in \mathbb{R}^{\mathcal{N}},$$

with $d_0^n = \Delta x Pe\left(\frac{\lambda}{2}+\nu\right)\chi_{[0,t_{\text{in}}]}(t^n)$, $\lambda = \frac{\Delta t}{\Delta x}$, $\nu = \frac{\Delta t}{Pe\Delta x^2}$, $e_1 = (1,0,\ldots,0)^T \in \mathbb{R}^{\mathcal{N}}$, and $h_{zj}^n = h_z(c_{aj}^n, c_{bj}^n, q_{zj}^n) = \frac{L}{Q/(\epsilon A_c)}\kappa_z\left(f_z(c_{aj}^n, c_{bj}^n) - q_{zj}^n\right)$, $j = 1,\ldots,\mathcal{N}$. Here, the function f_z is defined in (2.2). The parameter μ characterizes the operating conditions, e.g., $\mu := (Q, t_{\text{in}})$.

2.2.2 Numerical discretization of the SMB model

As mentioned earlier, the main dynamics of the fluid flow in all columns in an SMB unit are the same except for the node balance relations. We first address the discretization of one chromatographic column and then assemble the systems to obtain the discrete system for the whole SMB unit.

As done for batch chromatography earlier, we use the same strategy to discrete the SMB model. That is, we use the finite volume method to discretize the model in the kth column of the SMB unit (2.4)–(2.7), where the Lax–Friedrichs flux [111] is applied to solve the convection flux, and the central difference scheme is applied to evaluate the diffusion flux and the Crank–Nicolson scheme for the temporal discretization.

Let $\Delta t = 1/K$, and $t^n = n\Delta t, n \in \mathbb{K} := \{0,1,\ldots,K\}$ be the $K+1$ time instants over the time interval $[0,1]$, $\Delta x = 1/\tilde{\mathcal{N}}$ be the spatial grid size, and $c_{z,k}^n(\mu), q_{z,k}^n(\mu) \in \mathbb{R}^{\tilde{\mathcal{N}}}$ be the numerical approximations of the concentrations ($c_{z,k}$ and $q_{z,k}$, respectively) in the kth column at time instance $t = t^n$. Note that the time interval $[0,1]$ is actually a dimensionless switching period. The fully discretized finite volume formulation for the kth chromatographic column can be written as

$$A_k(\mu)c_{z,k}^{n+1}(\mu) = B_k(\mu)c_{z,k}^n(\mu) + r_{z,k}^n(c_{z,k}^{\text{in}}(t^n),\mu) - \frac{1-\epsilon}{\epsilon}\Delta t h_{z,k}^n(\mu), \tag{2.16}$$

$$q_{z,k}^{n+1}(\mu) = q_{z,k}^n(\mu) + \Delta t h_{z,k}^n(\mu) \tag{2.17}$$

where $A_k(\mu), B_k(\mu) \in \mathbb{R}^{\tilde{\mathcal{N}}\times\tilde{\mathcal{N}}}$ are tridiagonal matrices,

$$h_{z,k}^n(\mu) := f_z(c_{a,k}^n(\mu), c_{b,k}^n(\mu)) - q_{z,k}^n(\mu) \in \mathbb{R}^{\tilde{\mathcal{N}}},$$

is a nonlinear vector-valued function, and

$$r_{z,k}^n(c_{z,k}^{\mathrm{in}}(t^n),\mu) = r_{z,k}(c_{z,k}^{\mathrm{in}}(t^n),\mu)[1,0,\cdots,0]^T \in \mathbb{R}^{\tilde{\mathcal{N}}},$$

with

$$r_{z,k}(c_{z,k}^{\mathrm{in}}(t^n),\mu) := \left(1+\frac{Pe}{2\tilde{\mathcal{N}}}\right)\frac{\tilde{\mathcal{N}}Q_k(\mu)t_{\mathrm{s}}}{\epsilon L^2 A_c}c_{z,k}^{\mathrm{in}}(t^n)\Delta t \in \mathbb{R}. \tag{2.18}$$

The parameter μ characterizes the operating conditions of the underlying SMB process (e.g., $\mu = [Q_{\mathrm{I}},\ldots,Q_{\mathrm{IV}},t_{\mathrm{s}}]$).

It is noteworthy that $r_{z,k}^n(c_{z,k}^{\mathrm{in}}(t^n),\mu)$ in (2.16) depends on the information from the connected column, because the inflow of the kth column $c_{z,k}^{\mathrm{in}}(t^n)$ is determined by the outflow of the $(k-1)$th column, $k=2,\ldots,N_{\mathrm{col}}$, and $c_{z,1}^{\mathrm{in}}(t^n)$ is determined by $c_{z,N_{\mathrm{col}}}(t^n,1)$, the concentration at the outlet of the N_{col}th column, as detailed in (2.8)–(2.12) in the previous subsection. Thus, the system (2.16)–(2.17) are coupled with the systems corresponding to the other columns. Assembling all the systems, we have the FOM for the SMB unit as follows:

$$A(\mu)c_z^{n+1}(\mu) = B(\mu)c_z^n(\mu) + r_z^n(\mu) - \frac{1-\epsilon}{\epsilon}\Delta t h_z^n(\mu), \tag{2.19}$$

$$q_z^{n+1}(\mu) = q_z^n(\mu) + \Delta t h_z^n(\mu), \tag{2.20}$$

where

$$c_z^n(\mu) = \begin{bmatrix} c_{z,1}^n(\mu) \\ \vdots \\ c_{z,N_{\mathrm{col}}}^n(\mu) \end{bmatrix}, \quad q_z^n(\mu) = \begin{bmatrix} q_{z,1}^n(\mu) \\ \vdots \\ q_{z,N_{\mathrm{col}}}^n(\mu) \end{bmatrix}, \quad h_z^n(\mu) = \begin{bmatrix} h_{z,1}^n(\mu) \\ \vdots \\ h_{z,N_{\mathrm{col}}}^n(\mu) \end{bmatrix} \in \mathbb{R}^{\mathcal{N}},$$

$A(\mu) = \mathrm{diag}(A_1(\mu),\ldots,A_{N_{\mathrm{col}}}(\mu)) \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $B(\mu) = \tilde{B}(\mu) + B_r(\mu)$ with a block diagonal matrix $\tilde{B}(\mu) = \mathrm{diag}(B_1(\mu),\ldots,B_{N_{\mathrm{col}}}(\mu)) \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ and a (fairly) sparse matrix $B_r(\mu)$ resulting from separating an auxiliary vector

$$r_z^n(c_z^{\mathrm{in}}(t^n),\mu) := \begin{bmatrix} r_{z,1}^n(c_{z,1}^{\mathrm{in}}(t^n),\mu) \\ \vdots \\ r_{z,N_{\mathrm{col}}}^n(c_{z,N_{\mathrm{col}}}^{\mathrm{in}}(t^n),\mu) \end{bmatrix} = B_r(\mu)c_z^n(\mu) + r_z^n(\mu)$$

into two parts depending on the definition of $r_{z,k}(c_{z,k}^{\mathrm{in}}(t^n),\mu)$ in (2.18), $B_r(\mu) \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $r_z^n(\mu) \in \mathbb{R}^{\mathcal{N}}$, and $\mathcal{N} = \tilde{\mathcal{N}}\cdot N_{\mathrm{col}}$. The first part, $B_r(\mu)c_z^n(\mu)$, linearly depends on $c_z^n(\mu)$ and is determined by (2.8)–(2.12). The second part, $r_z^n(\mu)$, only depends on the parameter μ,

which leads to an efficient implementation using the ROM, to be introduced in Chapter 6. Nevertheless, this does not introduce any extra cost for the FOM simulation.

As mentioned earlier, the regime of the SMB is a CSS, which is characterized by an identical transient concentration profiles during two consecutive switching periods. Mathematically, given an initial state, the system (2.19)–(2.20) is solved step by step within a switching period; at the end of a period the state $(c_z^K(\mu),\, q_z^K(\mu))$ undergoes a shift, and the shifted vectors $(P_{\mathrm{s}}c_z^K(\mu), P_{\mathrm{s}}q_z^K(\mu))$ are used as the new initial state to continue the evolution process until the CSS conditions are satisfied. Here P_{s} is a permutation matrix. To determine the condition of the CSS, the following two criteria can be used:

1. check whether the concentrations at the beginning of two consecutive periods are identical, i.e.,

$$\max_{z\in\{a,b\}} \max\{\|c_z^0(\mu) - P_{\mathrm{s}}c_z^K(\mu)\|, \|q_z^0(\mu) - P_{\mathrm{s}}q_z^K(\mu)\|\} < \varepsilon_{\mathrm{CSS}},$$

   where $\varepsilon_{\mathrm{CSS}}$ is a user-specified tolerance;

2. check whether the outputs of interest, e.g., the purity of products (to be defined in Chapter 6), in two consecutive periods are equal.

2.3 Optimization of chromatography

The optimal operation of chromatography is of practical importance since it allows to exploit the full economic potential of the process and reduce the separation cost. During the past years, many efforts have been made in this area. For optimization of batch chromatography, the early work can be found in an extensive review [78] and references therein. An iterative optimization approach is addressed in [67], and a hierarchical approach on optimal control for a hybrid batch chromatographic process was developed in [74]. Optimization of SMB chromatography can be found, e.g., in [11, 12, 51, 112, 174, 50, 113, 104, 152, 168, 173]. However, almost all these studies are based on full-order model. Such a model with a large number of DOFs is able to capture the dynamics of the process, and the accuracy of the optimal solution obtained from that can be guaranteed. On the other hand, solving the FOM-based optimization is usually expensive. Thus, we explore the framework of surrogate-based optimization accelerated by reduced-order modeling. We

will first review three commonly used surrogate model based optimization methods and then focus on ROM based optimization.

The optimization problem of chromatography can be formulated as follows:

$$\begin{aligned} & \min_{\mu\in\mathcal{P}} \left\{ \mathcal{J}\left(u(t,x;\mu);\mu\right)\right\}, \\ \text{s.t.} \quad & \Psi\left(u(t,x;\mu);\mu\right) \leq 0, \\ & \Phi\left(u(t,x;\mu);\mu\right) = 0, \end{aligned} \tag{2.21}$$

where $\mathcal{J}$ is the objective function, Ψ defines the inequality constraints. The field variable $u(t,x;\mu)$ is the solution to the underlying parameterized PDEs $\Phi(u(t,x;\mu);\mu) = 0$, and the optimal variable μ is composed of the operating conditions. In this particular applications, $\mathcal{J}$ refers to the productivity or the throughput, Ψ refers to certain process constraints, $\Phi(\cdot)$ refers to the systems (2.1)–(2.3) for batch chromatography, and the systems (2.4)–(2.1.2) for SMB chromatography, respectively. Details will be given in Chapter 6.

2.3.1 High-fidelity model based optimization

In practical computation, the PDEs are usually discretized such that the optimization problem in (2.21) is replaced by an optimization problem in finite dimensions:

$$\begin{aligned} & \min_{\mu\in\mathcal{P}} \left\{ \tilde{\mathcal{J}}(u^{\mathcal{N}}(t;\mu);\mu)\right\}, \\ \text{s.t.} \quad & \tilde{\Psi}\left(u^{\mathcal{N}}(t;\mu);\mu\right) \leq 0, \\ & \tilde{\Phi}\left(u^{\mathcal{N}}(t;\mu);\mu\right) = 0, \end{aligned} \tag{2.22}$$

where $u^{\mathcal{N}} := u^{\mathcal{N}}(t;\mu) \in \mathbb{R}^{\mathcal{N}}$ is the solution to the discretized system of equations $\tilde{\Phi}\left(u^{\mathcal{N}}(t;\mu);\mu\right) = 0$, and $\tilde{\mathcal{J}}, \tilde{\Psi}$, and $\tilde{\Phi}$ are the operators in the finite dimensional vector space corresponding to $\mathcal{J}, \Psi$, and Φ, respectively. The discretized equations are often of large scale and complex. At each iteration of the optimization process, such a large-scale complex system of equations must be solved at least once to compute the objective and/or constraints. Actually, many studies on optimization of chromatography as mentioned earlier are based on the finely discretized FOM. Such a model with a large number of DOFs is able to capture the dynamics of the process, and the accuracy of the optimal solution obtained from that can be guaranteed. However, the expensive FOM must be repeatedly solved in the optimization process, which makes the runtime for obtaining the

optimal solution rather too long. To accelerate the underlying optimization, a surrogate ROM has been employed to replace the original large-scale discretized system for a rapid evaluation of the vector $u^{\mathcal{N}}$.

2.3.2 Surrogate model based optimization

The optimization with constraints including partial differential equations (PDE constrained optimization, for short), has emerged as a challenging research area in the last decades. It has arisen from various contexts, such as optimal design, control, and parameter estimation. Over the past years, besides the increasing progress of the computing hardware, many attempts have been made to develop efficient algorithms and strategies for solving such optimization problems; see, e.g., [29, 30, 93] and references therein. To accelerate an optimization process, the use of surrogate models has been gained increasing popularity during the past decades [1, 2, 18, 28, 186]. Roughly speaking, three kinds of models are often used as surrogate models: data fits, lower-fidelity models, and reduced-order models [55, 56].

Data fit type surrogate models, e.g., Kriging models and models using radial basis functions, are constructed via interpolation or regression of a set of input-output statistic quantities from the original model [139, 169]. The physics of the original system are entirely ignored in the construction of the surrogate such that the data fit surrogate models are considered as nonphysical-based approximations. As a result, it may cause unacceptable approximation errors when the physics of the system are strongly input-dependent [40].

Lower-fidelity models are derived from high-fidelity models by using such as coarser discretization grids (in space and/or time), relaxed solver tolerance, or by neglecting physics from the original models. They are physic-based surrogates because the main physics are still kept, unlike the data fit models. Moreover, it is very easy to generate the surrogate based on the high-fidelity model. However, the speedup by using the lower-fidelity surrogate is usually moderate, though it is cheaper than the original one [112]. With no doubt, aggressively neglecting physics or coarsening the grid often results in poor quality of the approximation.

Reduced-order models are constructed based on the high-fidelity models via various MOR methods, and ROM-based optimization has been extensively studied by mathematicians

and engineers [5, 8, 48, 58, 186]. Although the construction of ROMs can be fairly intensive, the speedup is usually significant, and the main characteristics of the original system can still be retained by the ROMs. Some comparisons between the aforementioned surrogate models for a statistical inverse problem can be found in [64], and a general discussion on these surrogate models can be found in [26].

Concerning the complex dynamics and nonlinearity involved in the chromatographic process, we pursue reduced-order modeling for batch and SMB chromatography in this thesis. Till now, applications of (P)MOR to chromatography are very limited in the literature. A POD method is employed for reduced-order modeling of a nonlinear SMB model in [112], and a Krylov subspace MOR method is applied to a linear SMB model in [114]. Nevertheless, these two methods are non-parametric MOR methods. It means that the resulting ROMs are accurate only locally, i.e., it is reliable only in the neighborhood of the parameter at which the ROM is constructed. Usually, the ROM needs to be updated during the optimization process.

In the following chapters, we will show the idea of (P)MOR methods and how the methods are applied to chromatographic models. Our goal is to use PMOR to construct a single ROM that meets the accuracy requirements over a wide range of the parameter domain. Given any feasible parameter, the output response can be rapidly obtained based on the ROM. As a result, when the FOM is replaced by the ROM during the optimization process, the optimal solution can be obtained within significantly reduced runtime.

Chapter 3

Model Order Reduction

Contents

In this chapter, we address the methodology of MOR. We begin with a brief review of (P)MOR in general and then focus on the RBM for parameterized nonlinear evolution problems. For parametric systems, the construction of a ROM and the simulation based on the ROM are often realized by the strategy of offline-online decomposition, as mentioned in Section 1.2. For efficient offline-online computations, techniques for dealing with the nonlinearity and/or nonaffinity are discussed. In particular, the empirical interpolation

method is reviewed in detail. Different basis construction methods are discussed. Finally, the offline-online decomposition technique is summarized. Parts of the contents in this chapter were originally presented in [188, 189].

3.1 Review of (P)MOR methods

(P)MOR has been proved to be a useful tool in handling large-scale computations. As mentioned in Chapter 1, various (P)MOR methods have been developed over the past years. In this section, we give a compact review of these commonly used (P)MOR methods. For conciseness, we show the basic idea and the main features of these methods. We introduce the method adopted in this thesis to deal with the motivating examples introduced in Chapter 2. Moreover, we also point out some typical issues on which we aim to focus in the following chapters.

Roughly speaking, (P)MOR can be classified into frequency-domain methods and time-domain methods. The former include BT methods and Krylov subspace methods. The latter mainly refer to the time-domain snapshot-based MOR methods, e.g., POD methods and RBMs.

3.1.1 Projection framework of MOR

To illustrate the basic idea of MOR, we consider a special case of (1.3), a non-parametric linear time-invariant (LTI) system,

$$E\dot{u}(t) = Au(t) + Bp(t), \tag{3.1}$$

$$y(t) = Cu(t), \tag{3.2}$$

where $E, A \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $B \in \mathbb{R}^{\mathcal{N}\times n_{\mathrm{I}}}$, $C \in \mathbb{R}^{n_{\mathrm{O}}\times\mathcal{N}}$ are constant matrices, $u(t) \in \mathbb{R}^{\mathcal{N}}$ is the state vector, $p(t) \in \mathbb{R}^{n_{\mathrm{I}}}$ is the input, and $y(t) \in \mathbb{R}^{n_{\mathrm{O}}}$ is the output. Often, $\mathcal{N}$, the order of the LTI system, is very large, and $n_{\mathrm{I}}, n_{\mathrm{O}} \ll \mathcal{N}$. The fundamental observation is that the solution to the LTI system often resides in a (relatively) low dimensional subspace of $\mathbb{R}^{\mathcal{N}}$. MOR aims at constructing a ROM that can reproduce the main input-output characteristics of the original system in (3.1)–(3.2). For all projection-based MOR

methods,[1] a right projection (reduced basis) matrix $V \in \mathbb{R}^{\mathcal{N}\times N}$, whose columns span a basis of the subspace where u can be well represented, is computed. A left projection matrix $W \in \mathbb{R}^{\mathcal{N}\times N}$ is constructed based on proper approximation principles. The ROM is obtained by using the approximation $u(t) \approx \hat{u}(t) := Vu_r(t)$ and employing Petrov–Galerkin projection with W,

$$\hat{E}\dot{u}_r(t) = \hat{A}u_r(t) + \hat{B}p(t), \tag{3.3}$$

$$\hat{y}(t) = \hat{C}u_r(t), \tag{3.4}$$

where $\hat{E} = W^T EV \in \mathbb{R}^{N\times N}$, $\hat{A} = W^T AV \in \mathbb{R}^{N\times N}$, $\hat{B} = W^T B \in \mathbb{R}^{N\times n_{\mathrm{I}}}$, $\hat{C} = CV \in \mathbb{R}^{n_{\mathrm{O}}\times N}$ are the reduced matrices, $u_r(t) \in \mathbb{R}^N$ is the reduced state vector, and $N \ll \mathcal{N}$. In contrast to the FOM in (3.1)–(3.2), the ROM in (3.3)–(3.4) is much cheaper to solve so that the state vector $u(t)$ and the output $y(t)$ can be rapidly recovered by $\hat{u}(t)$ and $\hat{y}(t)$, respectively. Various MOR methods have been developed during the past years, and they differ in the construction of the projection matrices V and W.

3.1.2 Frequency-domain MOR methods

At first, we introduce the transfer function of the system (3.1)–(3.2), which plays crucial roles in the frequency-domain methods. Let $U(s), P(s)$, and $Y(s)$ be the Laplace transforms of $u(t), p(t)$, and $y(t)$, respectively. Assuming $u(0) = 0$, the LTI system (3.1)–(3.2) (in the time domain) is transferred to an algebraic system of equations (in the frequency domain) as below:

$$sE\,U(s) = A\,U(s) + BP(s), \tag{3.5}$$

$$Y(s) = CU(s). \tag{3.6}$$

The transfer function $H(s)$ is defined as

$$H(s) = Y(s)/P(s) = C(sE - A)^{-1}B. \tag{3.7}$$

It is worth noting that the transfer function characterizes the relationship between the output ($Y(s)$) and the input ($P(s)$) in the frequency domain. Accordingly, the transfer

[1] For problems that are not explicitly described by a system of equations and the only access to the dynamics is via input/output measurements, non-projection MOR methods have been developed; see, e.g., [99, 110]. It is beyond the scope of this thesis.

function for the ROM is

$$\hat{H}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B}. \tag{3.8}$$

Moreover, the error between the transfer function of the FOM and that of the ROM,

$$\|H(\cdot) - \hat{H}(\cdot)\|, \tag{3.9}$$

is typically used to measure the accuracy of the ROM by frequency-domain MOR methods.

Balanced truncation methods

In the systems and control theory community, balanced truncation (BT) [125] is one of the most popular techniques for approximating linear dynamical systems. BT seeks for a state representation of the original system such that the representative states are both well controllable and well observable, meanwhile, those states that are least controllable and least observable are truncated. To this end, the starting point of the BT method is to quantify the controllability and observability of the states $u(t) \in \mathbb{R}^{\mathcal{N}}$. Usually, two associated generalized Lyapunov equations

$$A\mathbf{P}E^T + E\mathbf{P}A^T + BB^T = 0, \tag{3.10}$$

$$A^T\mathbf{Q}E + E^T\mathbf{Q}A + C^TC = 0, \tag{3.11}$$

need to be solved to determine the so-called controllability Gramian $\mathbf{P}$ and observability Gramian $\mathbf{Q}$, based on which the reduced-order system can be derived. For example, given the Cholesky factorizations $\mathbf{P} = Z_P Z_P^T$ and $\mathbf{Q} = Z_Q Z_Q^T$, let

$$Z_P^T Z_Q = [Z_1 \ Z_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix} =: Z\Sigma Y^T, \tag{3.12}$$

be an SVD of $Z_P^T Z_Q$ with $\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_N)$, $\sigma_N \ll \sigma_{N+1}$, $N < \mathcal{N}$. Then, the matrices V and W can be formulated as follows:

$$V = Z_P Z_1 \Sigma^{-1/2}, \quad W = Z_Q Y_1 \Sigma^{-1/2} \ \in \mathbb{R}^{\mathcal{N}\times N}. \tag{3.13}$$

Note that the diagonals of Σ, $\sigma_1, \sigma_2, \ldots, \sigma_{\mathcal{N}}$, are the Hankel singular values (HSVs), which play important roles in BT, because they provide a measure of controllability and observability for each state in the system [125]. The states corresponding to the smallest HSVs are least controllable and observable. Thus, these states could be ignored. In fact, constructing a ROM by retaining the states corresponding to the largest HSVs yields not only

a balanced truncation but also a global error bound between the transfer function of the ROM and that of the FOM [9, 68], i.e.,

$$\|H - \hat{H}\|_{\mathcal{H}_\infty} \leq 2(\sigma_{N+1} + \cdots + \sigma_{\mathcal{N}}), \tag{3.14}$$

where N is the order of the ROM, and $\sigma_{N+1}, \ldots, \sigma_{\mathcal{N}}$ are the neglected HSVs. The $\mathcal{H}$-∞ norm $\|\cdot\|_{\mathcal{H}_\infty}$ is defined by

$$\|H\|_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(H(\jmath\omega)),$$

where $\jmath = \sqrt{-1}$ and $\sigma_{\max}(H(\jmath\omega))$ is the largest singular value of the matrix $H(\jmath\omega)$. The computable global error bound in (3.14) allows an adaptive selection of the order of the ROM according to a user-specified error tolerance, which is an advantage of the BT method. Another advantage is that asymptotic stability is preserved by the reduced-order system.

As mentioned previously, the BT method relies on solving the associated (generalized) Lyapunov equations in (3.10)–(3.11) to construct the ROM, which is usually a computational bottleneck of this method. In fact, the computational cost of solving a (generalized) Lyapunov equation increases exponentially with respect to the order of the original system. At present, using efficient algorithms on advanced computers, really large-scale (generalized) Lyapunov equations can be solved within a reasonable amount of time [19]. The BT method has been extended to solve descriptor systems (i.e., E in (3.1) is singular); see, e.g., [89, 123, 155]. BT used in the framework of PMOR can be found in [20, 26].

Krylov subspace methods

To show the basic idea of Krylov subspace methods, we first recall two basic concepts in linear algebra, namely, Krylov subspace and block Krylov subspace. Given a matrix $L \in \mathbb{C}^{\mathcal{N}\times\mathcal{N}}$ and a vector $r \in \mathbb{C}^{\mathcal{N}}$, the Krylov subspace $\mathcal{K}_j(L, r)$ is defined as

$$\mathcal{K}_j(L, r) = \text{span}\,\{r, Lr, L^2, \ldots, L^{j-1}r\}. \tag{3.15}$$

Here, the integer j is called the order of the Krylov subspace. The block Krylov subspace $\mathcal{K}_j(L, R)$ is defined as

$$\mathcal{K}_j(L, R) = \text{span}\,\{R, LR, L^2R, \ldots, L^{j-1}R\}. \tag{3.16}$$

where $R = [r_1, \ldots, r_k] \in \mathbb{C}^{\mathcal{N}\times k}$ has more than one columns. Note that the dimension of the Krylov subspace $\mathcal{K}_j(L, R)$ can be smaller than the number of the column vectors, i.e., $k \cdot j$.

We now expand the transfer function defined in (3.7) at an expansion point s_0 as follows:

$$\begin{aligned} H(s) &= C[(s - s_0 + s_0)E - A]^{-1}B \\ &= C[(s_0E - A) + (s - s_0)E]^{-1}B \\ &= C[I + (s - s_0)(s_0E - A)^{-1}E]^{-1}(s_0E - A)^{-1}B \\ &= \sum_{i=0}^{\infty} \underbrace{C[-(s_0E - A)^{-1}E]^i(s_0E - A)^{-1}B}_{=:m_i(s_0)}(s - s_0)^i. \end{aligned} \tag{3.17}$$

Here, $m_i(s_0)$ is called the ith moment of the transfer function about s_0, $i = 0, 1, \ldots$. In particular, for $s = \infty$, the moments are also called Markov parameters, which can be computed by $C(E^{-1}A)^{j-1}E^{-1}B$ [19].

The projection matrices V and W are computed from the vectors associated with the moments, e.g.,

$$\begin{aligned} \text{range}(V) &= \text{colspan}\{\tilde{B}(s_0), \tilde{A}_B(s_0)\tilde{B}(s_0), \ldots, (\tilde{A}_B(s_0))^{j-1}\tilde{B}(s_0)\} =: \mathcal{K}_j(\tilde{A}_B(s_0), \tilde{B}(s_0)), \\ \text{range}(W) &= \text{colspan}\{\tilde{C}(s_0), \tilde{A}_C(s_0)\tilde{C}(s_0), \ldots, (\tilde{A}_C(s_0))^{j-1}\tilde{C}(s_0)\} =: \mathcal{K}_j(\tilde{A}_C(s_0), \tilde{C}(s_0)), \end{aligned}$$

where

$$\begin{aligned} \tilde{A}_B(s_0) &= (A - s_0E)^{-1}E, & \tilde{B}(s_0) &= (A - s_0E)^{-1}B, \\ \tilde{A}_C(s_0) &= (A - s_0E)^{-T}E^T, & \tilde{C}(s_0) &= (A - s_0E)^{-T}C^T, \end{aligned}$$

and $j \ll \mathcal{N}$ [73]. That is, V and W are taken as the basis vectors of the Krylov subspaces $\mathcal{K}_j(\tilde{A}_B(s_0), \tilde{B}(s_0))$ and $\mathcal{K}_j(\tilde{A}_C(s_0), \tilde{C}(s_0))$, respectively. Note that this process can be efficiently implemented by the Arnoldi algorithm or the Lanczos procedure [65]. Krylov subspace methods construct ROMs in such a way that first moments of the transfer function $H(s)$ are matched by those of the transfer function $\hat{H}(s)$ of the ROM, i.e., $m_i(s_0) = \hat{m}_i(s_0)$, $i = 0, \ldots, 2j - 1$ [73].

The ROM constructed via moment-matching methods with a single expansion point is reliable only locally. In fact, using a single expansion point, the accuracy of the resulting ROM depends on the order of the Krylov subspace adopted in the construction of the

projection matrices. To improve the accuracy of the approximation over a wide range of parameter values, multi-point moment-matching methods, also called rational Krylov methods, have been developed [19].

Note that the Krylov subspace methods can usually be efficiently implemented for large-scale sparse systems in MEMS simulations and integrated circuit design applications. However, stability cannot be preserved by the ROM in general. In [135], it has been proved that stability can be preserved when the system matrices satisfy certain conditions, see also [63]. A drawback of the Krylov subspace methods is that the *a priori* global error bound is not generally available, though much progress has been made in recent years; see, e.g., [14, 138, 137]. More recently, some *a posteriori* error bounds are proposed for (non-)parametric linear systems [24, 62]. For more details about Krylov subspace methods, e.g., the theoretical analysis and applications, please refer to [9, 14, 19, 63, 65, 73].

3.1.3 Time-domain MOR methods

The time-domain methods discussed here mainly refer to snapshot-based MOR methods, namely, POD methods and RBMs. These methods construct the projection matrices through extracting information from the snapshots in the time domain. The snapshots are taken from the solutions of the FOM at different parameter samples and/or different time instances (for time-dependent problems). Thus, these methods can start with a semi-discretized system (1.1) or a fully discretized system (1.2). Moreover, in the time-domain MOR methods, the accuracy of the resulting ROM is usually measured by, e.g., the error of the field variable $\|u - \hat{u}\|$ or that for the output $\|y - \hat{y}\|$.

Proper orthogonal decomposition

The proper orthogonal decomposition (POD) is a well-established tool for data analysis and data compression and is often used to construct an orthogonal basis (called POD basis) for MOR of linear and nonlinear dynamical systems. POD was introduced for the analysis of turbulence by Lumley in [117] and is also known as the principle component analysis in statistical analysis [96, 102], the Karhunen-Loève expansion in stochastic process modeling [107, 115], and empirical orthogonal eigenfunctions in atmospheric modeling [134].

The POD basis vectors are computed from the snapshots, which can be obtained form

the simulation of the FOM. Let $\{u_1, \ldots, u_{n_\mathrm{s}}\} \subset \mathbb{R}^{\mathcal{N}}$ be a set of snapshots, which are the solutions to the underlying system at different parameter samples and/or different time instances. POD is closely related to SVD of rectangular matrices in a finite-dimensional space. To compute a POD basis, define the snapshot matrix $U \in \mathbb{R}^{\mathcal{N}\times n_\mathrm{s}}$ whose jth column is the snapshot u_j, i.e., $U = [u_1, \ldots, u_{n_\mathrm{s}}]$. The POD basis vectors can be obtained via an SVD of the snapshot matrix U. Assume that an SVD of U is written as

$$U = Q\Sigma Y^T, \tag{3.18}$$

where the columns of $Q \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ and $Y \in \mathbb{R}^{n_\mathrm{s}\times n_\mathrm{s}}$ are the left and right singular vectors of U, respectively, $\Sigma \in \mathbb{R}^{\mathcal{N}\times n_\mathrm{s}}$ is a rectangular diagonal matrix with non-negative real numbers on the diagonal. Let $n_{\min} = \min(\mathcal{N}, n_\mathrm{s})$, and the diagonal entries σ_i $(i = 1, 2, \ldots, n_{\min})$ of Σ are called the singular values of the matrix U. Note that Q and Y are orthogonal matrices, i.e., $Q^TQ = I_{\mathcal{N}}$ and $Y^TY = I_{n_\mathrm{s}}$. Here, I_m refers to the identity matrix in $\mathbb{R}^{m\times m}$, $m = \mathcal{N}$, n_s. The matrix V of the POD basis with dimension k is defined as the k left singular vectors corresponding to the leading k singular values.

The number of the POD basis vectors, k, is typically chosen as the smallest integer such that

$$\frac{\sum_{j=k+1}^{n_{\min}} \sigma_j^2}{\sum_{j=1}^{n_{\min}} \sigma_j^2} \leq \epsilon_{\mathrm{POD}}, \tag{3.19}$$

where ϵ_{POD} is a user-specified tolerance, usually taken as 0.1% or smaller. In applications to fluid dynamics, the dominant POD basis vectors correspond to the most energetic flow modes in the system. The POD basis with dimension determined by (3.19) is often interpreted as that it captures $100(1 - \epsilon_{\mathrm{POD}})\%$ "energy" of the original system. It has been widely used for linear and nonlinear problems in various applications.

Note that the POD basis has the following interesting properties:

• The POD basis is not unique, due to the non-uniqueness of the SVD of a matrix.

• The POD basis V with dimension k is "optimal" in the sense that it minimizes the least square error of the reconstructed approximation of the snapshots. That is, for any k dimensional basis $Z \in \mathbb{R}^{\mathcal{N}\times k}$, we have

$$\min_{Z\in\mathbb{R}^{\mathcal{N}\times k}} \|U - ZZ^TU\|_F^2 = \|U - VV^TU\|_F^2 = \sum_{j=1}^{n_\mathrm{s}} \|u_j - VV^Tu_j\|_2^2 = \sum_{j=k+1}^{n_{\min}} \sigma_j^2. \tag{3.20}$$

Here $\|\cdot\|_F$ is the Frobenius norm of a matrix. It is noteworthy that the optimal approximation ($VV^T u_j$) for the snapshot u_j in (3.20) does not guarantee that the reduced approximation $\hat{u} := Vu_r$ is optimal, where u_r is obtained from the ROM based on the POD basis V [38]. Therefore, the error expression in (3.20) does not applied to the resulting ROM and yields no direct information regarding the accuracy of the reduced approximation $\hat{u}$.

As pointed out in [44], when the spatial dimension $\mathcal{N}$ of the discretization is much larger than the number of snapshots n_s, i.e., $\mathcal{N} \gg n_\mathrm{s}$, it may not be efficient to apply SVD directly on U. Instead, the eigenvalue decomposition of smaller matrix $U^T U \in \mathbb{R}^{n_\mathrm{s} \times n_\mathrm{s}}$ can be employed to efficiently compute the POD basis. In addition, several methods for computing a POD basis are summarized in [40]. On the other hand, if the number of snapshots is much larger than the spatial dimension $\mathcal{N}$, i.e., $n_\mathrm{s} \gg \mathcal{N}$, then it is costly to compute the POD basis based on the snapshot matrix U. This may happen when a very large number of time steps are needed to capture the dynamics. In such a case, there are a lot of redundant and linearly dependent information in the trajectory. The ASS technique can be employed as a preprocessing to filter out many redundant vectors from the original snapshot candidates, which will be further addressed in Chapter 5.

It should be pointed out that using the standard POD (i.e., projecting the original large-scale system onto the subspace spanned by the POD basis) alone for nonlinear problems may not yield vast reduction, because the complexity and the computational cost of the nonlinear terms cannot be reduced by projection. Further reduction techniques, like the (discrete) empirical interpolation method [17, 43] (to be further addressed in Section 3.4) can be combined to produce more efficient ROMs. In addition, POD is more often used for problems with no parameter dependence, though it has extended to parametric systems [46, 79, 142, 143]. For parametric systems, the RBM has gained increasing popularity, which is addressed as follows.

Reduced basis method (RBM)

The RBM is a useful PMOR method for parameterized PDEs, and it is implemented in the time domain. Here, we give a compact review of the development of the RBM and highlight its main features. We show the framework of RBMs for parameterized nonlinear evolution equations in detail in Section 3.2.

The RBM was introduced in the late of 1970s for nonlinear structural analysis [3, 133], and then it was further extended to various applications such as the incompressible Navier-Stokes equations in fluid dynamics [100, 144]. In particular, the last decade has witnessed tremendous development of the RBM for problems described by parameterized PDEs. The RBM has been developed for linear elliptic coercive and non-coercive equations [90, 91, 97, 140, 147, 158, 178], linear parabolic equations [72, 69, 85], quadratically nonlinear elliptic or parabolic equations [70, 106, 132, 176, 177, 187, 188], and nonlinear hyperbolic equations [47, 84]. Like the POD method discussed earlier, the RBM is also a snapshot-based MOR method, i.e., the RB is built upon the snapshots. In fact, most of the applications mentioned earlier are based on the Lagrange approach, i.e., only the snapshots are used to construct the basis. Nevertheless, one may also consider the Taylor [133, 146] or the Hermite [100] approach. In the Taylor approach, the basis is computed based on the snapshot (solution) and its N derivatives w.r.t. the parameter at a certain parameter value, while in the Hermite approach, the basis is computed based on the snapshots and their first derivatives w.r.t. the parameter at several parameter values. Along with the remarkable development during the past years, the following three aspects are emphasized in the RBM:

(i) efficient *a posteriori* error estimation for the output or the field variable, which is used to guide the parameter sampling during the basis extension process and to quantify the ROM for any given parameter online;

(ii) parameter sampling technique, which is crucial for effectively collecting system information over a wide range of parameters to construct an "optimal" RB in an affordable amount of time, especially for problems with high dimension of the parameter space;

(iii) offline-online decomposition, which decouples the construction of a ROM and the simulation based on the ROM. This permits a large upfront process of precomputation in the offline stage to build a simulation-efficient ROM, which is used to obtain a rapid response for any given input parameter in the online stage.

Note that in the RBM, the RB is usually constructed iteratively via a greedy algorithm [178]. When a sharp error bound or the true error is used in the greedy algorithm, the dimension of the RB can be kept as small as possible for a given error tolerance for the

ROM. As mentioned in Chapter 1, the output error bound for nonlinear time-dependent problems is less explored; see, e.g., [49, 132]. This motivates us to derive some efficient error bounds for these problems, which will be further addressed in Chapter 4.

It should also be pointed out that the offline computation can be fairly expensive because many repeated runs have to be done based on the (large-scale) FOM. To reduce the offline cost, we will introduce two accelerating techniques in Chapter 5. Although the RBM is often formulated in functional space or in weak formulation of a PDE, e.g., in the finite element space [140], it can also be derived algebraically [149, 148]. We will show the RBM in the vector space for parameterized nonlinear evolution equations in the next section.

The RBM is widely used in heat transfer, fluid dynamics, solid mechanics, electromagnetics, chemical engineering, even finance, etc. Nowadays, the RBM has become an important tool to solve parameterized PDEs. The RBM is built upon the traditional numerical discretization methods, e.g., the finite element method, the finite volume method, or the finite difference method. Usually, it is measured against those methods in the following sense: (a) the RB is actually built by the snapshots, which are the simulation data from the FOM obtained from one of those discretization methods; and (b) the RB solution does not directly approximate the exact (analytical) solution of the PDE, but a faithful numerical solution. This also applies to the POD method. For more details about the RBM and *a posteriori* error estimation, we refer to [81, 140, 150, 148].

3.1.4 A brief comparison

BT and Krylov subspace methods are initially devised for LTI systems arising from systems and control theory or circuit simulation. Both methods are usually implemented in the frequency domain, while POD and RBMs are implemented in the time domain. It should be noted that POD can also be implemented in the frequency domain [105, 160, 181]. A time-domain Krylov subspace based MOR method is presented in [54]. BT and Krylov subspace methods have been extended to linear time-varying systems and/or weakly nonlinear problems. In short, two main strategies were proposed to deal with the nonlinearity in the original system, i.e., approximation of the nonlinear function by polynomials of low degree [15, 145, 156, 157], and transformation of the original system into a quadratic bilinear system [23, 75, 76]. In addition, BT and Krylov subspace methods have also been

extended to parametric systems [26]. For a systematic discussion on the two frequency-domain methods, we refer to recent reviews in [19, 63].

POD and the RBM are widely applied to linear and nonlinear problems in a broad variety of applications. There are slight differences between POD and the RBM, though both are snapshot-based methods and implemented in the time domain. It is more often that POD is used for problems with no parameter dependence, although POD has been used in the parameter domain [35, 118] and has also been extended to problems that depend on both time and parameters [46, 79]. Moreover, the error measured in the two methods are different. POD aims at minimizing the SVD approximation error, as shown in (3.20), while the RBM aims at minimizing the reduced approximation error through a greedy algorithm [81], as will be further addressed in Section 3.3.

As mentioned earlier, both POD and RBMs are snapshot-based methods. This means that simulation based on the FOM needs to be performed, possibly at many parameter samples, to construct the ROM, which may result in intensive offline computations. By contrast, the frequency-domain MOR methods discussed earlier do not need to solve the FOM in the time domain to construct the ROM. As a consequence, the frequency-domain MOR methods are independent of the input $p(t)$, which commonly exists in the models from circuit or MEMS design. Nevertheless, the time-domain MOR methods are highly dependent on the input, and thus representative training inputs must be carefully selected to generate the snapshots. However, many systems from applications have only constant inputs, e.g., some from chemical engineering, which can easily exempt from selecting the training inputs.

To sum up, the study of MOR for linear, non-parametric problems has reached a considerable level of maturity, reflected by many survey papers and books [9, 10, 22, 19, 26, 27, 140, 149, 167]. However, MOR for parametric dynamical systems from many applications has attracted increasing attention. In these applications, the parametric system needs to be simulated many times, and the computation should be done preferably in a limited amount of time. Therefore, a ROM that is reliable over a wide range of parameter values is desired. This drives the development of PMOR methods.

Concerning the motivating examples introduced in Chapter 2, which are described by parameterized time-dependent nonlinear PDEs, we choose the RBM as a tool to handle the large-scale computations. In the next section, We will show the RBM in the vector

space for a fully discrete system resulting from parameterized nonlinear evolution equations and highlight some issues to be addressed in the following chapters.

3.2 PMOR for parameterized evolution equations via the RBM

In this section, we consider a class of evolution equations exemplified by the models introduced in Chapter 2. We show the construction of the ROM, the idea of PMOR based on projection and the issue of simulating the ROM.

3.2.1 Parametric nonlinear systems

We now consider a class of problems described by a parameterized evolution equations as follows:

$$\partial_t u(t,x;\mu) + \mathcal{L}(\mu)[u(t,x;\mu)] = 0, \quad t \in (0,T], \quad x \in \Omega \subset \mathbb{R}^d, \quad \mu \in \mathcal{P} \subset \mathbb{R}^p, \tag{3.21}$$

where $\mathcal{L}(\mu)[\cdot]$ is a spatial differential operator. This is a general expression of an evolution equation, e.g., the time-dependent convection-diffusion equation in (2.4). For discretization, let $0 = t^0 < t^1 < \cdots < t^K = T$ be $K+1$ time instants over the time interval $[0,T]$, and $\mathcal{W}^{\mathcal{N}} \subset L^2(\Omega)$ be an $\mathcal{N}$-dimensional discrete space in which an approximate numerical solution to (3.21) is sought. Given $\mu \in \mathcal{P}$ with suitable initial and boundary conditions, the numerical solution $u^n(\mu)$ at time $t = t^n$ can be obtained by using suitable numerical methods, e.g., the finite volume method. Assume that $u^n(\mu) \in \mathcal{W}^{\mathcal{N}}$ satisfies the fully discrete form

$$A_\mu^{(n)} u^{n+1}(\mu) = B_\mu^{(n)} u^n(\mu) + g(u^n(\mu);\mu), \tag{3.22}$$

where $A_\mu^{(n)}$, $B_\mu^{(n)} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ are the coefficient matrices at the time instance t^n, and $g(\cdot;\mu)$ is a nonlinear operator with respect to (w.r.t.) $u^n(\mu)$ and/or nonaffine w.r.t. the parameter μ. The superscript (n) and the subscript μ in $A_\mu^{(n)}$ and $B_\mu^{(n)}$ indicate the dependence on time and the parameter, respectively. The dimension $\mathcal{N}$ is usually large, implying that the numerical solution $u^n(\mu)$ is a faithful approximation and is often called the "true" solution. The resulting large-scale system in (3.22) is considered as a FOM. Often, the output

$$y^n(\mu) = l(u^n(\mu)), \tag{3.23}$$

is the quantity of interest.

Solving such a FOM repeatedly under parameter variations is time-consuming or even prohibitively costly in many-query contexts such as optimization, design, real-time control, and UQ.

3.2.2 Framework of PMOR

In many applications, it is observed that the solution to the parametric systems, $u(\mu)$, resides in a lower dimensional subspace $\mathcal{V}^N \subset \mathcal{W}^{\mathcal{N}}$, i.e., $u(\mu)$ can be well approximated by a properly chosen basis of the subspace. Similarly to the framework of (non-parametric) MOR presented Section 3.1.1, the idea of PMOR methods is that the underlying system of equations are projected onto a subspace spanned by a small number of properly chosen basis vectors via Petrov–Galerkin projection. Let $V, W \in \mathbb{R}^{\mathcal{N}\times N}$ be the projection matrices, and $\hat{u}^n(\mu) := V u_r^n(\mu)$ be the approximation of $u^n(\mu)$. The ROM

$$\hat{A}_\mu^{(n)} u_r^{n+1}(\mu) = \hat{B}_\mu^{(n)} u_r^n(\mu) + W^T g(V u_r^n(\mu); \mu) \tag{3.24}$$

that preserves the parameter μ as a symbol, should be sufficiently accurate for the variations of μ in the whole parameter domain. Here $\hat{A}_\mu^{(n)} = W^T A_\mu^{(n)} V \in \mathbb{R}^{N\times N}$, $\hat{B}_\mu^{(n)} = W^T B_\mu^{(n)} V \in \mathbb{R}^{N\times N}$ are the reduced matrices, and $u_r^n(\mu) \in \mathbb{R}^N$ is the vector of unknowns of the ROM.

Note that the ROM in (3.24) is derived using linear algebraic tools in a finite-dimensional vector space. It is also possible to formulate the ROM in a functional space using variational principle [69, 140, 147]. Since practical computations will be done in the discrete space, we will stick to the description of MOR in the vector space throughout this thesis. All the theories derived in this thesis are independent of the spatial discretization employed for the FOM.

Notably, the number of DOFs of the ROM in (3.24) is usually much smaller than that of the FOM in (3.22), i.e., $N \ll \mathcal{N}$. The goal of PMOR is that the ROM is much cheaper to solve compared to the FOM for any parameter μ. This is not necessarily achieved by (3.24); it is required that the evaluation of $\hat{A}_\mu^{(n)}$, $\hat{B}_\mu^{(n)}$ and $W^T g(V u_r^n(\mu); \mu)$ is done without resorting to the full dimension $\mathcal{N}$. For this, additional techniques may be necessary, as described in the following.

3.2.3 Simulation of the ROM

As mentioned earlier, the goal of PMOR is to provide a fast simulation stage, where for any given parameter μ the output response can be obtained rapidly based on the ROM. Particularly, in the RBM, an offline-online decomposition strategy is often employed to achieve this goal. In short, assume that the matrices $A_\mu^{(n)}$ and $B_\mu^{(n)}$ in (3.22) can be written in a separable way, the so-called affine form, i.e.,

$$A_\mu^{(n)} = \sum_{j=1}^{n_a} \xi_{\mu,j}^n A_j, \qquad B_\mu^{(n)} = \sum_{k=1}^{n_b} \zeta_{\mu,k}^n B_k, \tag{3.25}$$

where A_j, B_k are constant matrices, and $\xi_{\mu,j}^n$, $\zeta_{\mu,k}^n$ are the corresponding parameter- and time-dependent scalar coefficients. Note that the numbers n_a and n_b are desired to be small. Then

$$\hat{A}_\mu^{(n)} = W^T A_\mu^{(n)} V = \sum_{j=1}^{n_a} \xi_{\mu,j}^n \hat{A}_j, \qquad \hat{B}_\mu^{(n)} = W^T B_\mu^{(n)} V = \sum_{k=1}^{n_b} \zeta_{\mu,k}^n \hat{B}_k, \tag{3.26}$$

where $\hat{A}_j = W^T A_j V \in \mathbb{R}^{N\times N}$ and $\hat{B}_k = W^T B_k V \in \mathbb{R}^{N\times N}$, $j = 1, \ldots, n_a$, $k = 1, \ldots, n_b$. Note that once the projection matrices V and W are obtained, $\hat{A}_j$ and $\hat{B}_k$ can be precomputed, and in turn the evaluations of $\hat{A}_\mu^{(n)}$ and $\hat{B}_\mu^{(n)}$ at μ are independent of the full dimension $\mathcal{N}$. However, the computation of the last term of (3.24), $W^T g(V u_r^n(\mu); \mu)$, cannot be done analogously because of the nonlinearity or non-affinity of g. To achieve an efficient offline-online computation, a certain technique of interpolatory approximation can be applied, e.g., the empirical interpolation method (EIM) [17], the discrete empirical interpolation method (DEIM) [43], and the empirical operator interpolation method [87, 49]. In what follows, we briefly address the idea of EIM and discuss it in detail in Section 3.4.

To conduct an interpolatory approximation for the nonlinear part $g(\hat{u}^n(\mu), \mu)$ in (3.24), a parameter-independent basis $G \in \mathbb{R}^{\mathcal{N}\times M}$ ($M \ll \mathcal{N}$) is usually precomputed based on snapshots of the nonlinear function evaluations at a set of properly selected parameter samples. Then an affine approximation is defined by the interpolation operator $\mathcal{I}_M : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}}$, i.e.,

$$g(\hat{u}^n(\mu); \mu) \approx \mathcal{I}_M[g(\hat{u}^n(\mu); \mu)] := G\beta^n(\mu),$$

where $\beta^n(\mu) := \beta(u_r^n(\mu); \mu) \in \mathbb{R}^M$ is the corresponding vector of coefficients. The coefficient vector $\beta^n(\mu)$ can be determined, e.g., by the interpolation condition that $\hat{g}^n(\mu)$

interpolates $g(\hat{u}^n(\mu);\mu)$ at a set of properly selected components. As a result, a low dimensional ROM is obtained as

$$\hat{A}_\mu^{(n)} u_r^{n+1}(\mu) = \hat{B}_\mu^{(n)} u_r^n(\mu) + \hat{G}\beta^n(\mu), \tag{3.27}$$

where $\hat{G} = W^T G$ is precomputed. Given any feasible parameter value, the output response can be obtained rapidly because the computation is independent of the dimension $\mathcal{N}$ of the original FOM.

It is worth noting that the affine assumption in the coefficient matrices $A_\mu^{(n)}$ and $B_\mu^{(n)}$ in (3.22) can be relaxed for more general problems. That is, if the coefficient matrices have nonaffine parameter dependence, one can use certain approximation strategy to avoid costly $\mathcal{N}$-dependent evaluations in forming the reduced matrices. Please refer to [26, 128] for more details.

Now the question is how to compute the projection matrices V and W and the interpolation basis G, which is one of the key issues of PMOR. Recall that we formulate the ROM in (3.24) using Petrov–Galerkin projection, which is also called two-side projection, since V and W are different. In practice, Galerkin projection is often employed to construct the ROM, i.e., $W = V$, especially for elliptic problems. The reason is that the coefficient matrix, say, A, for simplicity, is symmetric positive definite for elliptic equations, and Galerkin projection can ensure the stability of the ROM. On the other hand, for parabolic or hyperbolic equations, Galerkin projection method is also practically employed for the sake of simplicity, as will be implemented in this thesis, although the stability cannot be guaranteed in general. Actually, the stability issue of the ROM is, in general, still an open problem, though many studies have been made and some strategies have been suggested to prevent producing unstable ROMs for certain problems [4, 16, 59, 91, 136, 154, 163, 183]. It is worth noting that for some time-independent problems in the form of $A(\mu)u = b(\mu)$, a Petrov–Galerkin projection method, which is also called the least-square projection, i.e., $W = AV$, has been employed [37, 91, 148, 165] to ensure the stability of the ROM.

In the RBM, the projection matrix V is also called the RB matrix, since the column vectors of V form a basis of the reduced subspace. Thus, we sometimes use RB matrix to refer to the projection matrix V in the following. Next, we first address the construction of the RB (matrix), and then discuss the generation of the interpolation basis for the nonlinear and/or nonaffine terms.

3.3 Construction of reduced basis

Construction of RB is one of the key issues for projection-based MOR, and various methods have been proposed in the past years. In this section, we mainly discuss the construction methods commonly used in the RBM.

For parametric systems, the RB is usually built iteratively through a greedy algorithm. In the following subsections, we first discuss the greedy algorithm, which can be used in general for basis construction, especially for PMOR methods. Then, we show a combination of the greedy algorithm and a POD procedure, namely, the POD-Greedy algorithm [85], which has been proved to be a very successful method for basis construction for parameter- and time-dependent problems.

3.3.1 Greedy algorithm

A greedy algorithm is a general procedure for solving complex and multi-step problems by making a locally optimal choice at each step in the hope that this choice will lead to a global optimal solution. It is widely used in various applications. In [178], the greedy algorithm was adopted to construct the RB in the RBM. Since then, this algorithm becomes very popular for recursively constructing RB in PMOR, especially in the RBM. Recently, it is also employed by other MOR methods, e.g., the multi-point moment-matching method [62], where the greedy algorithm is applied to choose the proper expansion points.

We now show the construction of the RB matrix V using a greedy algorithm. Generally, a training set $\mathcal{P}_{\text{train}}$ with a finite number of parameter samples is chosen *a priori* as a surrogate of the admissible parameter space. Assume that $\psi_N(\cdot)$ is an error indicator for an approximation by the current RB with dimension N. At each extension step, a parameter $\mu_\star$, which causes the largest error measured by the error indicator $\psi_N(\cdot)$, is chosen from $\mathcal{P}_{\text{train}}$ to enrich the RB. This process continues until the accuracy requirement is satisfied, i.e., the error indicator goes below the user-specified error tolerance. In practice, we may also specify an integer $N_{\max}$ as the maximal number of basis vectors to stop the iteration in the case that the available error estimation is too rough. The greedy algorithm for RB construction is given in Algorithm 1.

Algorithm 1 Greedy algorithm

Input: $\mathcal{P}_{\text{train}}$, μ_0, $\varepsilon_{\text{ROM}}(<1)$, $N_{\max}$.

Output: RB $V = [v_1, \ldots, v_N]$.

1: Initialization: $N = 0$, $\mu_\star = \mu_0$, $\psi_N(\mu_\star) = 1$, $V = [\,]$.
2: **while** $\psi_N(\mu_\star) > \varepsilon_{\text{ROM}}$ & $N < N_{\max}$ **do**
3: Collect information from the chosen parameter $\mu_\star$ to enrich the RB V.
4: Update N.
5: Find $\mu_\star := \arg \max_{\mu \in \mathcal{P}_{\text{train}}} \psi_N(\mu)$.
6: **end while**

Note that the RB computed from the greedy algorithm is hierarchical in the sense that $\text{span}\{v_1, \ldots, v_{N-1}\} \subset \text{span}\{v_1, \ldots, v_N\}$ for $N = 2, \ldots, N_{\max}$, where $N_{\max}$ is the maximal dimension of the RB. This allows us to adjust the accuracy or the runtime of the ROM online by varying the dimension of the RB. In addition, the quality of the RB (or the ROM) depends crucially on two issues: the error indicator and the training set.

The error indicator $\psi_N(\cdot)$ can be the true error or a certain error bound/estimation for the reduced approximation against the reference quantity computed from the FOM. Because the true error requires the "true" solution by solving the original large-scale system, an error bound/estimation is preferable, which will be explored in Chapter 4.

The training set is a surrogate for the whole parameter domain. On the one hand, it should be sufficiently large so that more parameter information can be collected. On the other hand, its size has to be limited so that the computational cost is affordable. During the past years, many efforts have been devoted to seeking an "optimal" training set by certain adaptive techniques [34, 53, 82, 83, 141]. This will be further discussed in Chapter 5.

When Algorithm 1 is used for parameter-dependent steady problems, Step 3 performs a FOM simulation at $\mu_\star$ to acquire the solution $u(\mu_\star)$ and enriches RB V as $V = [V, u(\mu_\star)]$; when it is used for unsteady problems, Step 3 becomes more complicated, since for the chosen parameter $\mu_\star$, the solutions at many different time instances can be used to enrich the RB and a trivial procedure might result in loss of valuable information. In [85], POD and the greedy algorithm are combined as the POD-Greedy algorithm, which is often employed for the construction of RB for parameter- and time-dependent problems and will be addressed in detail in the next subsection.

3.3.2 POD-Greedy algorithm

The idea of the POD-Greedy algorithm [85] is to use the greedy procedure for the parameter sampling and POD of the snapshot matrix to compute the new basis vectors. Algorithm 2 shows the basic steps of the POD-Greedy algorithm.

Algorithm 2 POD-Greedy algorithm

Input: $\mathcal{P}_{\text{train}}$, μ_0, $\varepsilon_{\text{ROM}}(<1)$, $N_{\max}$.

Output: RB $V = [v_1, \ldots, v_N]$.

1: Initialization: $N = 0$, $\mu_\star = \mu_0$, $\psi_N(\mu_\star) = 1$, $\mathcal{V}_N = \{0\}$, $V = [\,]$.
2: **while** $\psi_N(\mu_\star) > \varepsilon_{\text{ROM}}$ & $N < N_{\max}$ **do**
3: Simulate the FOM at $\mu_\star$, and collect snapshots $\{u^n(\mu_\star)\}_{n=0}^{K}$.
4: Perform POD process:
Compute $\bar{U} := [u^0, \ldots, \bar{u}^K]$, $\bar{u}^n := u^n(\mu_\star) - \Pi_{\mathcal{V}_N}[u^n(\mu_\star)]$, $n \in \mathbb{K}$, where $\Pi_{\mathcal{V}_N}[\cdot]$ is the projection operator onto the current space $\mathcal{V}_N := \text{span}\{v_1, \ldots, v_N\}$.
Compute the first POD mode v_{N+1}, the left singular vector of the matrix $\bar{U}$.
5: Enrich the RB $V := [V, v_{N+1}]$.
6: Update $N = N + 1$.
7: Find $\mu_\star := \arg \max_{\mu \in \mathcal{P}_{\text{train}}} \psi_N(\mu)$.
8: **end while**

Remark 3.3.1. *The first POD mode refers to the first left singular vector which corresponds to the largest singular value of the matrix under consideration. More generally, one can enrich the current basis with more than one POD mode in each iteration [49, 132]. One may also directly apply POD to the snapshots ($u^n(\mu_\star)$ instead of $\bar{u}^n$) as in [132], then one additional orthogonalization process should be performed after each enrichment since the increased vectors can be linearly dependent on the current space.*

Remark 3.3.2. *For many problems, like the batch chromatographic model and the SMB model considered in this thesis, the total number of time steps in FOM simulation is very large. This implies that the number of snapshots K in Step 3 of Algorithm 2 is large if no appropriate pretreatment for the snapshots is applied. The large number of snapshots will result in expensive computations in Step 4. To tackle this problem, we propose a technique of adaptive snapshot selection to adaptively discard the redundant (linearly dependent) information from the trajectory, so the selected snapshots consist of only the most "representative" vectors from the trajectory and the runtime for the RB*

construction can be largely reduced. This will be detailed in Chapter 5.

Note that when the POD-Greedy algorithm is applied to time-dependent problems, one parameter might be repeatedly selected to enrich the current basis. This implies that some "useful" information corresponding to the parameter has not yet been fully used.

The study of *a priori* convergence of the greedy algorithm for the RB construction for steady problems can be found in [31, 33]. The extension to time-dependent problems, i.e., the convergence rate of the POD-Greedy algorithm is analyzed in [80]. It is shown that the approximation error of the RB generated by the (POD-)Greedy algorithm gives exponential or algebraic convergence under certain conditions.

As mentioned previously, both the POD method and the RBM are snapshot-based MOR methods. That is, the projection basis is built upon the snapshots. Thus, the quality of the snapshots is crucial for the accuracy of the final ROM. If the representative system information is not contained in the snapshots (solutions to the FOM at different time instances and/or different parameter samples in the training set), it is difficult or even impossible to construct an accurate ROM. An interesting study for the snapshot collection for POD is found in [98]. The authors show that the two different groups of snapshots, $\{u(\cdot, t^j) \mid j = 0, \ldots, K\}$ and $\{u(\cdot, t^j), u(\cdot, t^{j+1}) - u(\cdot, t^j), u(\cdot, t^K) \mid j = 0, \ldots, K-1\}$, may result in different POD bases. Similar discussion on the snapshots collection can also be found in the earlier work in [109]. These studies remind us that attention should be paid to the collection of snapshots when a snapshot-based MOR method is implemented.

For parametric systems, a single ROM that is reliable over the whole parameter domain is preferable. However, a single ROM might not be sufficient for a system that exhibits big differences as the parameters cross different sub-domains. For such a case, several adaptive techniques using local basis methods have been suggested [6, 7, 82, 103, 142]. In this thesis, we use a global basis for the underlying problems, i.e., only one ROM will be generated for each FOM. However, local basis methods might be considered in the future for nonlinear SMB chromatography with higher Péclet number to deal with its high complexity. More discussion on using local basis vs. global basis can be found in [26].

3.4 Empirical interpolation method and related topics

As mentioned earlier, if there are nonlinear and/or nonaffine operators in the FOM, the computational cost and complexity cannot be largely reduced by using projection, because the nonlinear and/or nonaffine part, e.g., $W^T g(V u_r^n(\mu); \mu)$ in (3.24), requires computation in the original high dimensional space. In such a case, a further efficient approximation of the nonlinear and/or nonaffine part is crucial for MOR. Otherwise, the reduction in computational time by MOR might be very limited. See some analysis in [43] for example.

During the last decade, many efforts have been made in dealing with the nonlinearity and/or nonaffinity. For systems that are nonlinear or with nonaffine parameter dependence, the interpolation technique, e.g., the empirical interpolation method (EIM) [17] or its variants, can be employed. The idea of these interpolation techniques is to construct an interpolant to approximate the nonlinear and/or nonaffine function.

Assume that we are given a nonaffinely parameter-dependent function $g(x;\mu)$, $(x;\mu) \in \Omega \times \mathcal{P} \subset \mathbb{R}^d \times \mathbb{R}^p$, with sufficient regularity, i.e., $g(\cdot;\mu) \in L^\infty(\Omega)$ for all $\mu \in \mathcal{P}$. Let

$$\mathcal{M}_g := \{g(x,\mu) \mid x \in \Omega, \mu \in \mathcal{P}\}, \tag{3.28}$$

be the manifold induced by the nonaffinely parameter-dependent function. The use of interpolatory approximation is based on the following observations [130]:

(i) limited parameter dependence: the manifold $\mathcal{M}_g$ is typically of low dimension, meaning that any element $g \in \mathcal{M}_g$ can be well represented by a few properly chosen (parameter-independent) basis functions; and

(ii) limited spatial dependence: although $g(x,\mu)$ is defined in the spatial domain Ω for all $\mu \in \mathcal{P}$, the spatial variation of g can be captured mainly by a set of small number of well "selected" points instead of arbitrary points in the entire spatial domain.

The first observation enables us to construct a parameter-independent basis with a preferably low dimension. The second one enables us to determine the interpolation coefficients at low cost, since for every parameter in the parameter domain the function values at only a few "selected" points are needed.

We now show the interpolation method in a general framework. Let $\{g_i(x)\}_{i=1}^M$ be a set of parameter-independent basis functions, which is usually computed based on the snapshots of the function evaluations at some carefully chosen parameter samples in the parameter domain, say, $\{g(\cdot,\mu_j) \mid \mu_j \in \mathcal{P}_g \subset \mathcal{P}\}$. The interpolant is defined as

$$g_M(x,\mu) := \sum_{i=1}^{M} g_i(x)\beta_i(\mu), \tag{3.29}$$

where $\beta_i(\mu) \in \mathbb{R}$, $i = 1\ldots,M$, are the corresponding parameter-dependent coefficients. To determine the coefficients, we need to choose a finite subset $\Omega_g := \{x_1^{\mathrm{EI}},\ldots,x_{n_g}^{\mathrm{EI}}\}$ $(\subset \Omega)$ on which $g_M(x,\mu)$ interpolates the exact values of $g(x,\mu)$.

Various algorithms have been proposed to conduct the interpolation. They differ in the way of computing the basis function and/or the interpolation points. For the basis construction, one may use a POD basis (e.g., in the DEIM [43] or the "best point" interpolation method (BPIM) [130]) or a iteratively constructed basis (e.g., in the EIM [17]). Note that a POD basis is not only linearly independent but also orthogonal. For the determination of the interpolation points, one may choose $n_g = M$, which is adopted by, e.g., the EIM, the DEIM, and the BPIM; or one may also choose $n_g > M$, which is adopted by, e.g., the Gauss-Newton with approximated tensor method [41].

We now address the implementation of the EIM in a finite-dimensional discrete space. The idea of EIM is to construct an affine expression $g_M(x;\mu)$ in (3.29) to approximate $g(x;\mu)$, i.e., $g(x,\mu) \approx g_M(x,\mu)$. The EIM determines the coefficients $\beta_i(\mu)$ by enforcing the interpolation condition that the approximation $g_M(x;\mu)$ interpolates the exact value of $g(x;\mu)$ at the empirical interpolation (EI) points $\Omega_g := \{x_1^{\mathrm{EI}},\ldots,x_M^{\mathrm{EI}}\}$, i.e.,

$$\sum_{i=1}^{M} g_i(x_j^{\mathrm{EI}})\sigma_i(\mu) = g(x_j^{\mathrm{EI}};\mu), \quad j = 1,\ldots,M. \tag{3.30}$$

In practical computations, the parameter-independent basis, called collateral reduced basis (CRB), and the EI points are usually computed in a finite-dimensional discrete space, say, $\mathbb{R}^{\mathcal{N}}$. Let $\mathcal{P}_{\mathrm{train}}^{\mathrm{CRB}} := \{\mu_j \mid \mu_j \in \mathcal{P}, j = 1,\ldots,n_{\mathrm{train}}\}$ be a training set with a finite number (n_{train}) of parameter samples, which is chosen as a surrogate of the parameter domain $\mathcal{P}$. Let $\mathcal{G} := \{g_1(\mu_j),\ldots,g_{n_{\mathrm{train}}}(\mu_j)\}$ be the set of snapshots, where $g(\mu_j) := [g(x_1,\mu_j),\ldots,g(x_{\mathcal{N}},\mu_j)]^T \in \mathbb{R}^{\mathcal{N}}$ is the vector of function evaluations of $g(x,\mu)$ at the parameter $\mu_j \in \mathcal{P}_{\mathrm{train}}^{\mathrm{CRB}}$ $(j = 1,\ldots,n_{\mathrm{train}})$ on the spatial grids $\{x_1,x_2,\ldots,x_{\mathcal{N}}\}$. Note that

in the discrete space, the basis G is actually a set of constant vectors, $g_1, \ldots, g_M \in \mathbb{R}^{\mathcal{N}}$, and the EI points $\{x_1^{\mathrm{EI}}, \ldots, x_M^{\mathrm{EI}}\} (\subset \{x_1 \ldots, x_{\mathcal{N}}\})$ are corresponding to a few indices in the spatial grids, denoted by $\{\wp_1, \ldots, \wp_M\} =: \mathcal{I}$. To use matrix-vector notations, we introduce a vector $\beta(\mu) = [\beta_1(\mu), \ldots, \beta_M(\mu)]^T \in \mathbb{R}^M$, and a matrix

$$S = [e_{\wp_1}, \ldots, e_{\wp_M}], \tag{3.31}$$

where $e_{\wp_i} = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^{\mathcal{N}}$ is the $\wp_i$th column of the identity matrix in $\mathbb{R}^{\mathcal{N}\times\mathcal{N}}$. For any $\mu \in \mathcal{P}$, let $g(\mu) = [g(x_1, \mu), \ldots, g(x_{\mathcal{N}}, \mu)]^T \in \mathbb{R}^{\mathcal{N}}$, and then the interpolation condition in (3.30) becomes

$$S^T G\beta(\mu) = S^T g(\mu). \tag{3.32}$$

Thus, $\beta(\mu) = (S^T W)^{-1} S^T g(\mu)$, and the interpolant defined by the EI basis W reads

$$\hat{g}(\mu) = G\beta(\mu) = G(S^T G)^{-1} S^T g(\mu), \quad \mu \in \mathcal{P}. \tag{3.33}$$

Note that $S^T G$ is invertible because it is actually a lower triangular matrix with unit diagonal elements [17, 71]. Given an error tolerance $\varepsilon_{\mathrm{CRB}}$, the procedure of constructing the CRB and the EI points is summarized in Algorithm 3.

Note that the framework of Algorithm 3 is in the vector space, and it is slightly different from the frameworks for constructing the CRB and the EI points in the literature, where they are mostly presented in the continuous functional/operator form, e.g., in [71, 70].

Remark 3.4.1. *For time-dependent problems, we do not treat time as a separate parameter, but put the time trajectory for all training samples together as the snapshots to construct the CRB. That is, in Algorithm 3, the input snapshot set is redefined as* $\mathcal{G} := \{g(u^k(\mu_j), \mu_j) \mid \mu_j \in \mathcal{P}_{train}^{CRB},\ j = 1, \ldots, n_{train};\ k = 0, 1, \ldots, K\}$. *Note that when the total number (*K*) of time steps for one FOM simulation is large, the number of snapshots will be* $(K+1) \cdot n_{train}$ *if no further snapshot selection strategy is employed, and this number can be huge. This may render the computation of CRB very expensive. Again, the adaptive snapshot selection [25, 187] can be employed to reduce the computational cost, analogously to the implementation for the RB construction; see Remark 3.3.2.*

This interpolation approximation serves to reduce the complexity in evaluation of the nonlinear and/or nonaffine parts of the model, which cannot be directly reduced by projection. It is crucial for an efficient offline-online computation for the RBM and other

Algorithm 3 Generation of CRB and EI points

Input: $\mathcal{G} := \{g(\mu_1), \ldots, g(\mu_{n_{\text{train}}})\}, \varepsilon_{\text{CRB}} < 1$.

Output: CRB $G = [g_1, \ldots, g_M]$ and EI points (indices) $\mathcal{I} := \{\wp_1, \ldots, \wp_M\}$.

1: Initialization: $G = [\,]$, $S = [\,]$, $\mathcal{I} = \emptyset$.

2: $m = 1$, $\xi_m = \arg\max_{g(\mu_j)\in\mathcal{G}} \|g(\mu_j)\|$, $\wp_m = \arg\max_{i\in\{1,\ldots,\mathcal{N}\}} |\xi_{m,i}|$, $g_m = \xi_m/\xi_{m,\wp_m}$, $G = [G, g_m]$, $S = [S, e_{\wp_m}]$, $\mathcal{I} = \mathcal{I} \cup \{\wp_m\}$.

3: **while** $\|\xi_m\| > \varepsilon_{\text{CRB}}$ **do**

4: $m = m + 1$.

5: For all $g(\mu_j) \in \mathcal{G}$, compute the interpolant $\hat{g}(\mu_j) = G(S^T G)^{-1} S^T g(\mu_j)$ (see (3.33)).

6: Define $g(\mu_m) := \arg\max_{g(\mu_j)\in\mathcal{G}} \|g(\mu_j) - \hat{g}(\mu_j)\|$ and the error $\xi_m := g(\mu_m) - \hat{g}(\mu_m)$.

7: **if** $\|\xi_m\| \le \varepsilon_{\text{CRB}}$ **then**

8: Stop and set $M = m - 1$.

9: **else**

10: Determine the next EI point and basis vector:

$$\wp_m = \arg\max_{i\in\{1,\ldots,\mathcal{N}\}} |\xi_{m,i}|, \quad g_m = \xi_m/\xi_{m,\wp_m}. \tag{3.34}$$

11: Update $G = [G, g_m]$, $S = [S, e_{\wp_m}]$, $\mathcal{I} = \mathcal{I} \cup \{\wp_m\}$.

12: **end if**

13: **end while**

MOR methods, to be discussed in Section 3.5. More details of the EIM, e.g., the error analysis and the applications to general nonlinear and/or nonaffine problems, can be found in [69, 70, 71, 52, 120]. Next, we review the related techniques for dealing with the nonlinearity and/or nonaffinity in MOR and highlight some remaining challenges for future work.

The BPIM is proposed in [130, 131]. The main difference between the EIM and the BPIM is the way of determining the interpolation points. The EIM defines the interpolation points by a greedy algorithm, as shown in Algorithm 3, while the BPIM defines them by solving an optimization problem. It is not surprising that the BPIM is usually more expensive than the EIM. Nevertheless, the performance of the EIM and the BPIM are pretty similar in many cases [66, 130].

The DEIM [43, 45] has gained increasing popularity for nonlinear MOR since it is easy to be implemented in many situations. The bases in the DEIM and the EIM are different, although they are both extracted from the snapshots of the nonlinear (or nonaffine) function evaluations at different time instances and/or different parameter samples. The basis in the DEIM is an orthogonal basis computed by POD, while the basis in the EIM is a basis iteratively determined by Algorithm 3 and it is not necessarily orthogonal. In addition, the ways of determining the interpolation points for the two methods are different. The interpolation points in the DEIM are determined based on the POD basis vectors and they are generated after the generation of the interpolation basis, while those in the EIM are determined based on the snapshots themselves and they are generated along with the interpolation basis. Note that the set of the snapshots might contain a very large number of vectors for some problems that depend on both parameter and time. This may make the computation of the POD basis very expensive or even prohibitive. It deserves further investigations in the future. Recently, a discrete matrix version of DEIM for nonaffine parameterized systems is proposed in [128].

For some applications, the nonlinear function might exhibit a wide range of behaviors as the parameter passes through different regimes of the parameter domain. In such a case, the dimension of the DEIM basis has to be taken very high to capture the main features over the whole parameter domain, which restricts the significance of the DEIM. More recently, a localized discrete empirical interpolation method (LDEIM) is proposed in [142]. The LDEIM constructs several local DEIM bases according to certain cluster or partition strategies in the offline stage and chooses one of them for the online simulation. Due to the small size of the local ROM, the reduction is expected to be more significant than that achieved by using a ROM based on a global DEIM basis. It is also worth noting that an adaptive method for online update of the DEIM basis via low-rank updates is presented in [143].

Recently, the EIM is extended to the empirical operator interpolation method (EOIM) for interpolation of operators [87, 49]. The EOIM is applicable to operators that depend on the field variable $u(t, x; \mu)$, e.g., $g(u(t, x; \mu), x; \mu)$. The evaluation of $g(x_j; \mu)$ in (3.30) is thus replaced by $g(u(t, x_j; \mu), x_j; \mu)$. In this thesis, we use empirical operator interpolation, where the nonaffine operator is in the form of $g(u(t, x; \mu); \mu)$.

More recently, the EIM [17] has been extended to the generalized empirical interpolation

method [119] in the sense that the evaluation at the interpolation points is replaced by a more practical evaluation at interpolating continuous linear functionals on a class of Banach spaces. Apart from the EIM type methods presented earlier, the missing point estimation [13] and the Gauss-Newton with approximated tensor method [41] are both based on the Gappy POD interpolation method [57] and are also used to deal with the nonlinear and nonaffine terms in many applications [13, 36, 42, 180].

Note that all the methods mentioned earlier share almost the same idea, namely, interpolating the nonlinear and/or nonaffine functions/vectors at a set of properly chosen points/entries with a few precomputed basis (functions). It is sometimes called "hyper reduction" in the literature [164]. A comprehensive review of the EIM and related approximation strategies mentioned earlier is found in [21].

The CRB is usually built before constructing the RB for an efficient offline-online computation. Here, one should be aware that the dimensions of the CRB and the RB should be well balanced in order to achieve a good reduction. In general, for fixed CRB, the accuracy of the ROM improves as the dimension of the RB increases; likewise, for fixed RB, the accuracy of the ROM improves as the dimension of the CRB is increases. However, in either case, the accuracy of the ROM cannot be further improved after the dimension of the basis reaches a certain stage because after certain iterations the approximation error caused by the fixed basis dominates the total error. In fact, this phenomenon has been reported in several studies [49, 71]. A simple solution is using a sufficiently high dimensional CRB to ensure the accuracy of the interpolation. Certainly, this is not the optimal way since the offline time could be unnecessarily large. In [49], a scheme of synchronized generation of both bases is proposed. The idea is as follows: a very low dimensional CRB is first built, and then the RB and the CRB are simultaneously enriched by the greedy algorithm if the error (measured by error bounds or true error) of the resulting ROM is decreased; otherwise, only the CRB is enriched. This process continues until the desired ROM is obtained. In this way, the dimensions for both bases could be well balanced. In general, efficiently treating nonlinearity and nonaffinity in MOR remains challenges, and it is still an active research area.

3.5 Offline-online decomposition

Typically, when PMOR methods are employed in the time-critical applications for which the model needs to be repeatedly simulated under parameter variations, the reduction is often realized by the strategy of offline-online decomposition. That is, all quantities that depend on the high dimension are precomputed and stored, e.g., on a supercomputer or in a parallel way. This process is the so-called offline stage, and it can be very expensive because the FOM computation is involved. However, it needs to be performed only once as preprocessing. For example, the parameter-independent components for the affine expression of the coefficient matrices in the ROM, i.e., $\hat{A}_j$ and $\hat{B}_k$ ($j = 1, \ldots, n_a$, $k = 1, \ldots, n_b$) in (3.26) and the parameter-independent reduced matrix $\hat{G}$ in (3.27) should be precomputed. In the online stage, for any given feasible value of parameters, the parameter-dependent reduced matrices, $\hat{A}_\mu^{(n)}$, and $\hat{B}_\mu^{(n)}$ in (3.27), can be rapidly assembled by using the precomputed data in the offline stage, and the output response can be cheaply obtained based on this small-size ROM. This process has no reference to the FOM, and the computation is only in the scale of the (low) dimension $N \ll \mathcal{N}$. Since the ROM is usually used many times under parameter variations in many-query contexts, the cost in the offline stage will be paid off.

To perform the offline-online decomposition, the affine assumption in the coefficient matrices $A_\mu^{(n)}$ and $B_\mu^{(n)}$ in (3.22) is often required for MOR of the underlying problem. In addition, the affine form is also important for an efficient computation during the offline phase when an error bound/estimation is employed for the greedy algorithm, which is to be further addressed in the next chapter.

Chapter 4

Output Error Bound and Estimation

Contents

In this chapter, we address the error control of the ROMs. To generate ROMs in a goal-oriented fashion, two output error bounds are proposed for parameterized nonlinear evolution equations. One is the primal-only output error bound, and the other is a primal-dual error bound. The performance of both error bounds is preliminarily demonstrated by two academic examples. Further applications to real-life models will be given in Chapter 6. This is one of the main contributions of this thesis. References [187, 188] originally presented this work.

4.1 Previous work and objectives

(P)MOR is aimed at constructing a simulation-efficient ROM to reproduce the dominant dynamics or the input-output response of the original large-scale system, at a compromise with accuracy to an acceptable extent. The question is how to efficiently measure the

error of the ROM. Obviously, computing the true error of the reduced approximation is not feasible because it requires the true solution from the FOM simulation, which is usually expensive. Instead, an efficient error estimation is desired to quantify the error caused by the ROM. In this chapter, we study *a posteriori* error bound/estimation for the ROMs.

As mentioned in Section 3.3.2, an *a posteriori* error estimation has gained much attention in RBMs over the past years. In fact, to generate a ROM, efficient *a posteriori* error estimation is crucial because it enables the generation to be reliable and automatic. More precisely, *a posteriori* error estimation plays important roles in both the offline and the online stages. As discussed in Section 3.3.1, in the offline stage, the use of an efficient error estimation (instead of the true error) permits sufficiently large training samples for the greedy algorithm to construct a reliable ROM within an affordable amount of time. In addition, a sharp error estimation enables us to build a sufficiently accurate ROM with its size as (comparably) small as possible. In the online stage, a rigorous error estimation can quantify the ROM for any given parameter. Rigorous, sharp, and cheaply computable are the desired properties of an efficient error estimation.

In the past years, many efforts have been devoted to the study of *a posteriori* error estimation for either the field variable (the solution to the underlying system) or the output of interest. Particularly, the RBM has a strong emphasis on the derivation of *a posteriori* error estimation. Research on an *a posteriori* error estimation for the RBM started with [133, 146] and has been followed by many others for different problems. Let us mention only a few previous work on *a posteriori* error estimation, for linear and affinely parameter-dependent problems [69, 72, 85, 147, 158, 159], for nonaffine and/or nonlinear problems [39, 49, 70, 129, 132, 176, 177], and related survey papers or monographs [140, 161, 162]. More recently, the space-time reduced basis method is introduced for linear or quadratically nonlinear parabolic problems [175, 183, 184, 185]. Notably, these error estimations are all derived in the functional space in the framework of the finite element discretization except for [49, 85]. In the finite element discretization framework, the weak form of the PDE is used to derive the error bound, while the error bound in [85] is derived in the framework of the finite volume discretization for error estimation of the field variables. Note that the finite element coercivity constant is usually time-consuming to compute, the so-called successive constraint method [97] is employed in practice. Other recent work of error bounds or estimations for POD-based ROMs can be found, e.g., in [45, 92, 94, 98, 153, 170, 182], which are usually derived either in continuous space or

based on finite difference discretization. It is worth noting that an *a priori* error estimation is derived for the use of the proper orthogonal decomposition technique in the context of option pricing models in [166]. For general linear dynamic systems, an *a posteriori* error estimation is proposed in [86].

In this chapter, we present two kinds of error estimations for projection-based PMOR methods applied to parameterized (nonlinear) evolution equations. One is a primal-only error bound and the other is a primal-dual error estimation. The first one is derived based on the analysis of the residual, and it shares similar ideas in [49] but it is derived in the vector space. This error bound is cheap to compute. Nevertheless, it may lose sharpness for convection dominated problems, especially when a large number of time steps are needed. The second one adopts a primal-dual approach, and it is also derived in the vector space. The derived error estimator is fairly sharp and takes a little bit more time to compute compared to the first one. Since the two error bounds are both derived algebraically in the vector space, they are independent of spatial discretization approach employed, and furthermore, they are applicable to any projection-based PMOR methods.

In what follows, the inner product is defined as $\langle z_1, z_2 \rangle := z_1^T z_2, \forall z_1, z_2 \in \mathbb{R}^{\mathcal{N}}$. The induced norm $\|\cdot\|$ is the standard 2-norm in the Euclidean space. However, if the discrete system of equations is obtained by using the finite element method, the solution to the discrete system is actually the coefficient vector corresponding to the basis vectors of the solution space. In such a case, the inner product should be defined properly with the mass matrix of the solution space, and the norm will be the corresponding induced norm.

4.2 A primal-only output error bound

As mentioned previously, it is crucial to derive a sharp, rigorous and inexpensive *a posteriori* error bound, which enables reliable and low-cost construction of the RB. One common technique for the derivation of the error estimator is based on the residual. In [49, 87], the authors provided an error estimation for the field variable in functional space for evolution equations, where an upper bound of the operator is employed for the error estimation due to the abstract expression in the functional space. Since all the simulations are done in the finite-dimensional vector space in practice, in this section, we derive an error estimation for the field variable directly in the vector space. Using the concrete expression in the

vector space, we can directly compute the norm of the operator for each parameter rather than use an upper bound. In this sense, the final error bound derived in the vector space will be sharper than that in the functional space. Moreover, we derive an output-oriented error bound based on the error estimation for the field variable. For many applications, the output response $y(u^{\mathcal{N}})$ is of interest. Hence, during the process of the greedy algorithm, e.g., Algorithm 2 or Algorithm 5, the error indicator $\psi_N(\mu_\star)$ should be the error estimation for the output response, such that the resulting ROM is expected to be more accurate and reliable.

For the parameterized evolution equation in (3.21), we derive an output error bound in the vector space for the ROM in (3.24). Recall that the evolution scheme in the vector space reads

$$A_\mu^{(n)} u^{n+1}(\mu) = B_\mu^{(n)} u^n(\mu) + g\left(u^n(\mu);\mu\right), \tag{4.1}$$

where $g\left(u^n(\mu);\mu\right) \in \mathbb{R}^{\mathcal{N}}$ is the nonlinear term, and $A_\mu^{(n)}$ and $B_\mu^{(n)}$ are nonsingular for a stable scheme in practice, $n = 0, \ldots, K-1$.

Let $\hat{u}^n(\mu) = V u_r^n(\mu)$ be the RB approximation of $u^n(\mu)$, and $\mathcal{I}_M : \mathbb{R}^{\mathcal{N}} \mapsto \mathbb{R}^{\mathcal{N}}$ be an interpolation operator, i.e., $\hat{g}(\hat{u}^n(\mu);\mu) := \mathcal{I}_M[g(\hat{u}^n(\mu);\mu)] = G\beta^n(\mu)$ is defined as the interpolant of the nonlinear term, where $V \in \mathbb{R}^{\mathcal{N}\times N}, G \in \mathbb{R}^{\mathcal{N}\times M}$ are the precomputed parameter-independent bases, $u_r^n(\mu) \in \mathbb{R}^N$, $\beta^n(\mu) \in \mathbb{R}^M$ are parameter-dependent coefficients. In the following, for the sake of simplicity, we omit the explicit expression of the dependence on μ in $u^n(\mu), \hat{u}^n(\mu), u_r^n(\mu)$ and $\beta^n(\mu)$, and use $u^n, \hat{u}^n, u_r^n$ and β^n instead. The following *a posteriori* error bound is based on the residual

$$r^{n+1}(\mu) := B_\mu^{(n)} \hat{u}^n + \mathcal{I}_M[g(\hat{u}^n;\mu)] - A_\mu^{(n)} \hat{u}^{n+1}. \tag{4.2}$$

With simple computations, we obtain the norm of the residual

$$\begin{aligned}
\left\|r^{n+1}(\mu)\right\|^2 &= \left\langle r^{n+1}(\mu), r^{n+1}(\mu)\right\rangle \\
&= (u_r^n)^T \underline{W^T (B_\mu^{(n)})^T B_\mu^{(n)} V} u_r^n + (\beta^n)^T \underline{S^T S} \beta^n \\
&\quad + \left(u_r^{n+1}\right)^T \underline{W^T (A_\mu^{(n)})^T A_\mu^{(n)} V} u_r^{n+1} + 2\left(\beta^n\right)^T \underline{S^T B_\mu^{(n)} V} u_r^n \\
&\quad - 2 (u_r^n)^T \underline{W^T (B_\mu^{(n)})^T A_\mu^{(n)} V} u_r^{n+1} - 2\left(\beta^n\right)^T \underline{S^T A_\mu^{(n)} V} u_r^{n+1}.
\end{aligned} \tag{4.3}$$

Based on the analysis of this residual, we have the error bound for the field variable as follows.

Proposition 4.2.1. *Assume that for all $\mu \in \mathcal{P}$ the operator $g(\cdot;\mu) : \mathcal{W}^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}}$ is Lipschitz continuous w.r.t. the first argument, i.e., there exists a positive constant L_g, such that*

$$\|g(x;\mu) - g(y;\mu)\| \leq L_g \|x - y\|, \quad x, y \in \mathcal{W}^{\mathcal{N}}, \quad \mu \in \mathcal{P},$$

and the interpolation of g is "exact" with a certain dimension of $G = [g_1, \ldots, g_{M+M'}]$, i.e.,

$$\mathcal{I}_{M+M'}\left[g(\hat{u}^n;\mu)\right] := \sum_{m=1}^{M+M'} g_m \beta_m^n(\mu) = g(\hat{u}^n;\mu).$$

Assume again, that for all $\mu \in \mathcal{P}$, the initial projection error is vanishing $e^0(\mu) = 0$, then the approximation error $e^n(\mu) := u^n - \hat{u}^n$ satisfies

$$\|e^1(\mu)\| \leq \mathtt{R}_\mu^{(0)}, \quad \|e^n(\mu)\| \leq \mathtt{R}_\mu^{(n-1)} + \sum_{k=0}^{n-2} \left(\prod_{j=k+1}^{n-1} \mathtt{G}_\mu^{(j)} \right) \mathtt{R}_\mu^{(k)}, \quad n = 2, \ldots, K, \tag{4.4}$$

where

$$\begin{aligned} \mathtt{R}_\mu^{(k)} &= \left\| (A_\mu^{(k)})^{-1} \right\| \left(\epsilon_{\mathrm{EI}}^k(\mu) + \|r^{k+1}(\mu)\| \right), \quad k = 0, \ldots, n-1, \\ \mathtt{G}_\mu^{(j)} &= \left\| (A_\mu^{(j)})^{-1} \right\| \left(\|B_\mu^{(j)}\| + L_g \right), \quad j = k+1, \ldots, n-1, \end{aligned}$$

and $\epsilon_{\mathrm{EI}}^n(\mu)$ is the error due to the EI, i.e.,

$$\epsilon_{\mathrm{EI}}^n(\mu) := g(\hat{u}^n;\mu) - \mathcal{I}_M[g(\hat{u}^n;\mu)] = \sum_{m=M+1}^{M+M'} \|g_m\| \cdot |\beta_m^n(\mu)| \,. \tag{4.5}$$

A sharper error bound can be given as

$$\begin{aligned} &\|e^1(\mu)\| \leq \eta_{N,M}^1(\mu) := \mathtt{R}_{\mathrm{F},\mu}^{(0)}, \\ &\|e^n(\mu)\| \leq \eta_{N,M}^n(\mu) := \mathtt{R}_{\mathrm{F},\mu}^{(n-1)} + \sum_{k=0}^{n-2} \left(\prod_{j=k+1}^{n-1} \mathtt{G}_{\mathrm{F},\mu}^{(j)} \right) \mathtt{R}_{\mathrm{F},\mu}^{(k)}, \quad n = 2, \ldots, K, \end{aligned} \tag{4.6}$$

where

$$\begin{aligned} \mathtt{R}_{\mathrm{F},\mu}^{(k)} &= \left\| (A_\mu^{(k)})^{-1} \right\| \epsilon_{\mathrm{EI}}^k(\mu) + \left\| (A_\mu^{(k)})^{-1} r^{k+1}(\mu) \right\|, \quad k = 0, \ldots, n-1, \\ \mathtt{G}_{\mathrm{F},\mu}^{(j)} &= \left\| (A_\mu^{(j)})^{-1} B_\mu^{(j)} \right\| + L_g \left\| (A_\mu^{(j)})^{-1} \right\|, \quad j = k+1, \ldots, n-1. \end{aligned}$$

Proof. By forming the difference between (4.1) and (4.2), we have the error equation

$$\begin{aligned} A_\mu^{(n)} e^{n+1}(\mu) &= B_\mu^{(n)} e^n(\mu) + g(u^n;\mu) - \mathcal{I}_M[g(\hat{u}^n;\mu)] + r^{n+1}(\mu) \\ &= B_\mu^{(n)} e^n(\mu) + (g(u^n;\mu) - g(\hat{u}^n;\mu)) + (g(\hat{u}^n;\mu) - \mathcal{I}_M[g(\hat{u}^n;\mu)]) + r^{n+1}(\mu). \end{aligned} \tag{4.7}$$

Left-multiplying on both sides of (4.7) with $(A_\mu^{(n)})^{-1}$, we obtain

$$\begin{aligned} e^{n+1}(\mu) =& (A_\mu^{(n)})^{-1} B_\mu^{(n)} e^n(\mu) + (A_\mu^{(n)})^{-1} \left(g(u^n;\mu) - g(\hat{u}^n;\mu)\right) \\ &+ (A_\mu^{(n)})^{-1} \left(g(\hat{u}^n;\mu) - \mathcal{I}_M\left[g(\hat{u}^n;\mu)\right]\right) + (A_\mu^{(n)})^{-1} r^{n+1}(\mu). \end{aligned} \tag{4.8}$$

Applying the Lipschitz condition of g, we have $\|g(u^n;\mu) - g(\hat{u}^n;\mu)\| \le L_g \|e^n(\mu)\|$. Then by the triangle inequality and the property of the matrix norm, we have

$$\left\|e^{n+1}(\mu)\right\| \le \left\|(A_\mu^{(n)})^{-1}\right\| \left(\|B_\mu^{(n)}\| + L_g\right) \|e^n(\mu)\| + \left\|(A_\mu^{(n)})^{-1}\right\| \left(\epsilon_{\mathrm{EI}}^n(\mu) + \left\|r^{n+1}(\mu)\right\|\right), \tag{4.9}$$

where $\epsilon_{\mathrm{EI}}^n(\mu)$ is the error due to the EI, as defined in (4.5). Resolving the recursion (4.9) with initial error $\left\|e^0(\mu)\right\| = 0$ yields the error bound in (4.4).

To obtain the error bound in (4.6), we re-observe the equation in (4.8) and see that the error bound in (4.9) is unnecessarily enlarged. A sharper bound for $\left\|e^{n+1}\right\|$ is given as

$$\begin{aligned} \left\|e^{n+1}(\mu)\right\| \le& \left(\left\|(A_\mu^{(n)})^{-1} B_\mu^{(n)}\right\| + L_g \left\|(A_\mu^{(n)})^{-1}\right\|\right) \|e^n(\mu)\| \\ &+ \left\|(A_\mu^{(n)})^{-1}\right\| \epsilon_{\mathrm{EI}}^n(\mu) + \left\|(A_\mu^{(n)})^{-1} r^{n+1}(\mu)\right\|, \end{aligned} \tag{4.10}$$

since the following two inequalities are true, i.e., $\left\|(A_\mu^{(n)})^{-1} B_\mu^{(n)}\right\| \le \left\|(A_\mu^{(n)})^{-1}\right\| \left\|B_\mu^{(n)}\right\|$ and $\left\|(A_\mu^{(n)})^{-1} r^{n+1}(\mu)\right\| \le \left\|(A_\mu^{(n)})^{-1}\right\| \|r^{n+1}(\mu)\|$. Resolving the recursion (4.10) with initial error $\left\|e^0(\mu)\right\| = 0$ yields the proposed error bound in (4.6). □

Remark 4.2.2. *In many cases, the matrices $A_\mu^{(n)}$ and $B_\mu^{(n)}$ in* (4.1) *are independent of t^n and μ; see, e.g., the batch chromatographic model considered in this thesis. In such a case, the error bound becomes much simpler, see* (6.11) *and* (6.13) *in Chapter 6.*

Remark 4.2.3. *In [49], the derivation of the error bound is based on the general operator form in the functional space. The error bound in* (4.4) *corresponds to the operator form (5.5) in [49]. However, the error bound may grow exponentially when $\mathtt{G}_\mu^{(j)} = \left\|(A_\mu^{(j)})^{-1}\right\| \left(\left\|B_\mu^{(j)}\right\| + L_g\right) > 1$ in* (4.4). *In the vector space, this problem can be easily avoided by using* (4.10) *instead of* (4.9) *if $\mathtt{G}_{\mathrm{F},\mu}^{(n)} = \left\|(A_\mu^{(n)})^{-1} B_\mu^{(n)}\right\| + L_g \left\|(A_\mu^{(n)})^{-1}\right\| \le 1$, whereby the sharper error bound in* (4.6) *is obtained.*

Remark 4.2.4. *For the computation of the error bound in* (4.4)*, we need to compute the norm of the residual* $r^{n+1}(\mu)$ *by using* (4.3)*. Note that all terms underlined in* (4.3) *can be efficiently computed once* V*,* W*, and* G *are obtained, once the affine expression of the coefficient matrices* $A_\mu^{(n)}$ *and* $B_\mu^{(n)}$ *is available. This is also true for the computation of* $\left\|(A_\mu^{(n)})^{-1}r^{n+1}(\mu)\right\|$ *for the error bound in* (4.6)*. Consequently, the evaluation of the error bound is cheap due to its independence of* $\mathcal{N}$*. In addition, as is shown in [49], small* M' *gives good results in practice; we use* $M' = 1$ *in the latter simulations.*

In many applications, the quantity of interest is not the field variable itself, but some outputs. In such a case, it is desired to estimate the output error in order to construct a goal-oriented ROM. Based on the error estimation for the field variable above, we have the output error estimation in the following.

Proposition 4.2.5. *Under the assumptions of Proposition 4.2.1, assume the output of interest,* $y\left(u^n(\mu)\right)$*, can be expressed in the following form:*

$$y(u^n(\mu)) = Pu^n, \tag{4.11}$$

where $P \in \mathbb{R}^{N_\mathrm{O}\times\mathcal{N}}$ *is a constant matrix, then the output error* $e_\mathrm{O}^n(\mu) := Pu^n - P\hat{u}^n$ *satisfies*

$$\begin{aligned}\left\|e_\mathrm{O}^{n+1}(\mu)\right\| &\le \tilde{\eta}_{N,M}^{n+1}(\mu)\\ &:= \mathtt{G}_{\mathrm{O},\mu}^{(n)}\eta_{N,M}^n + \left\|P(A_\mu^{(n)})^{-1}\right\|\epsilon_\mathrm{EI}^n(\mu) + \|P\|\left\|(A_\mu^{(n)})^{-1}r^{n+1}(\mu)\right\|,\end{aligned} \tag{4.12}$$

where $\mathtt{G}_{\mathrm{O},\mu}^{(n)} = \left\|P(A_\mu^{(n)})^{-1}B_\mu^{(n)}\right\| + L_g\left\|P(A_\mu^{(n)})^{-1}\right\|$*,* $n = 0,\ldots,K-1$*.*

Proof. Left-multiplying on both sides of the error equation (4.8) with P, we obtain

$$\begin{aligned}Pe^{n+1}(\mu) = P\big((A_\mu^{(n)})^{-1}B_\mu^{(n)}e^n(\mu) + (A_\mu^{(n)})^{-1}\left(g(u^n;\mu) - g\left(\hat{u}^n;\mu\right)\right)\\ + (A_\mu^{(n)})^{-1}(g(\hat{u}^n;\mu) - \mathcal{I}_M[g(\hat{u}^n;\mu)]) + (A_\mu^{(n)})^{-1}r^{n+1}(\mu)\big).\end{aligned}$$

Applying the Lipschitz condition of g and using the triangle inequality, as well as the property of the matrix norm, we have

$$\begin{aligned}\left\|e_\mathrm{O}^{n+1}(\mu)\right\| &= \left\|Pe^{n+1}(\mu)\right\|\\ &\le \mathtt{G}_{\mathrm{O},\mu}^{(n)}\left\|e^n(\mu)\right\| + \left\|P(A_\mu^{(n)})^{-1}\right\|\epsilon_\mathrm{EI}^n(\mu) + \|P\|\left\|(A_\mu^{(n)})^{-1}r^{n+1}(\mu)\right\|.\end{aligned} \tag{4.13}$$

Replacing $\|e^n(\mu)\|$ in (4.13) with its bound in (4.6), we obtain the proposed output error bound in (4.12). □

Remark 4.2.6. *Once the error estimation for the field variable is obtained, e.g.,* (4.6)*, a trivial error bound for the output* (4.11) *can be given as*

$$\begin{aligned}\left\|e_{\mathrm{O}}^{n+1}(\mu)\right\| &= \left\|Pe^{n+1}(\mu)\right\| \\ &\leq \|P\| \left\|e^{n+1}(\mu)\right\| \\ &\leq \|P\| \left(\mathtt{G}_{\mathrm{F},\mu}^{(n)} \left\|e^{n}(\mu)\right\| + \left\|(A_{\mu}^{(n)})^{-1}\right\| \epsilon_{\mathrm{EI}}^{n}(\mu) + \left\|(A_{\mu}^{(n)})^{-1} r^{n+1}(\mu)\right\|\right).\end{aligned} \tag{4.14}$$

The last inequality is true due to the inequality (4.10)*. It is obvious that the bound for* $\left\|e_{\mathrm{O}}^{n+1}(\mu)\right\|$ *in* (4.13) *is sharper than that in* (4.14)*. As a result, the final output error bound in* (4.12) *is sharper than the trivial output error bound derived in* (4.14)*.*

Note that the error bound for the field variable $\eta_{N,M}^{n}(\mu)$ is involved in the output error bound $\tilde{\eta}_{N,M}^{n+1}(\mu)$. Moreover, the former is a summation of the residual and the error $\epsilon_{\mathrm{EI}}^{n}(\mu)$ over all the previous time steps. This implies that both error bounds are accumulated over time. As a result, they may lose sharpness when a large number of time steps are needed, e.g. in the simulation of batch chromatography [187]. The same phenomenon also exists in the error estimation in [49]. Similar observations are reported in [132]. To circumvent the problem, we propose a sharper output error bound for the ROM in the next section.

4.3 A primal-dual output error bound

For (nonlinear) evolution equations, time-stepping schemes are often used to solve them [126], and error estimations for projection-based MOR methods have been studied in recent years; see, e.g., [72, 69, 85]. Most of the existing error estimators in the literature may tend to lose sharpness when a large number of time steps are needed, because the error estimators are actually a summation of the error over the previous time steps. To circumvent this problem, we introduce a suitable dual system at each time instance in the evolution process associated with the primal system, i.e., the original system. The output error for the primal system can thus be estimated sharply and efficiently with the help of the dual system.

Actually, an *a posteriori* output error bound for the RBM using the primal-dual approach can be found in [72], where the derived output error bound is derived for linear evolution equations. From the numerical comparison in Section 4.4, it is shown that the proposed error estimation outperforms the error bound in [72] for a linear evolution system.

The aforementioned error bounds introduced in [175, 183, 184, 185] are based on space-time variational formulation. Notably, the space-time error bounds are derived for the space-time model rather than the standard model addressed in many other MOR papers. The space-time model is obtained from the parabolic equations by first discretizing in space and then integrating in time. As a result, the state vector (unknown vector) of the space-time model is different from the state vector for the standard model derived by only discretization in space, and time-stepping in time. Roughly speaking, the solution vector of the space-time model can be considered as a long vector $u_{\mathrm{st}} \in \mathbb{R}^{\mathcal{N}\cdot K}$ including the spatial discretized vector at all the time steps, where $\mathcal{N}$ is the number of spatial grids and K is the number of time steps. The corresponding error bound measures the error of this long vector computed by the corresponding ROM. Our error bound is defined for the solution vector $u \in \mathbb{R}^{\mathcal{N}}$ of the spatially discretized model at each time step t^k, $k = 1, \ldots, K$. The errors measured by the error bounds are different. Finally, the error bounds in those papers are valid only for linear and at most quadratically nonlinear systems. The error bounds are limited to Petrov–Galerkin discretization (in space), and an inf-sup constant (or its lower bound) for the corresponding variational (weak) form must be available. Our proposed error bound is valid for general linear and nonlinear systems (given Lipschitz continuity of the nonlinear term) and is applicable to any discretization approach.

The idea for the proposed error estimation originates from the recent study in [61, 62], where some error bounds are derived for linear time-invariant systems. The main difference of the proposed error estimation from that in [61, 62] is that the new error estimation is derived directly in the time domain and is exactly designed for the output in the time domain. It is particularly useful for snapshot-based MOR methods, e.g., the RBM [85, 140, 187] and the proper orthogonal decomposition (POD) method [35, 43, 180, 181]. It is valid for nonlinear parametric systems, whereas the error bound in [61, 62] is an error estimation for the transfer function of the ROM, so that it is used for linear parametric systems. In other words, it is an error estimation for the output in the frequency domain, which is well suited for the frequency-domain MOR methods, e.g., the Krylov subspace method [14, 24, 65].

In the following, we introduce the proposed error estimation for the output error in the time domain. Assume that the FOM from the spatial and temporal discretization of the PDEs can be written as

$$A_{\mu}^{(n)} u^{n+1}(\mu) = b(u^n(\mu); \mu), \tag{4.15}$$

where $A_\mu^{(n)}$ is assumed to be nonsingular for all $\mu \in \mathcal{P}$, $u^n(\mu) \in \mathcal{W}^{\mathcal{N}}$ is the numerical solution at time $t = t^n$, $b(\cdot;\mu) : \mathcal{W}^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}}$ can be nonlinear (or linear) w.r.t. the first argument and/or nonaffine w.r.t. the parameter $\mu \in \mathcal{P}$, e.g., the right-hand side of the equation in (3.22). The output of interest is expressed as in (4.11). Here, we temporally assume $N_{\mathrm{O}} = 1$ for simplicity. The extension to the multiple output case is straightforward; see Remark 4.3.8.

To derive an efficient output error estimation, at each time step, we denote the original system as the primal system

$$\begin{cases} A_\mu^{(n)} u^{n+1}(\mu) = b(u^n(\mu);\mu), \\ \qquad y^{n+1}(\mu) = P u^{n+1}(\mu) \end{cases} \tag{4.16}$$

and introduce a corresponding dual system as follows:

$$(A_\mu^{(n)})^T u_{\mathrm{du}}^{n+1}(\mu) = -P^T. \tag{4.17}$$

Assume that $(V_{\mathrm{pr}}, W_{\mathrm{pr}})$ and $(V_{\mathrm{du}}, W_{\mathrm{du}})$ are the projection matrix pairs for MOR of the primal and dual systems, respectively. Using Petrov–Galerkin projection, we have the ROMs for the primal and the dual systems, respectively,

$$\begin{cases} W_{\mathrm{pr}}^T A_\mu^{(n)} \hat{u}^{n+1}(\mu) = W_{\mathrm{pr}}^T b(\hat{u}^n(\mu);\mu), \\ \qquad \hat{y}^{n+1}(\mu) = P \hat{u}^{n+1}(\mu), \end{cases} \tag{4.18}$$

$$W_{\mathrm{du}}^T (A_\mu^{(n)})^T \hat{u}_{\mathrm{du}}^{n+1}(\mu) = -W_{\mathrm{du}}^T P^T, \tag{4.19}$$

where $\hat{u}^n(\mu) = V_{\mathrm{pr}} a_{\mathrm{pr}}^n(\mu)$, $\hat{u}_{\mathrm{du}}^n(\mu) = V_{\mathrm{du}} a_{\mathrm{du}}^n(\mu)$ are the approximations to $u^n(\mu)$ and $u_{\mathrm{du}}^n(\mu)$, respectively. The vectors $a_{\mathrm{pr}}^n(\mu)$ and $a_{\mathrm{du}}^n(\mu)$ are the unknowns of the reduced primal and the reduced dual systems in (4.18) and (4.19), respectively. The residuals for both systems read

$$r_{\mathrm{pr}}^{n+1} := r_{\mathrm{pr}}^{n+1}(\mu) = b(\hat{u}^n(\mu);\mu) - A_\mu^{(n)} \hat{u}^{n+1}(\mu), \tag{4.20}$$

$$r_{\mathrm{du}}^{n+1} := r_{\mathrm{du}}^{n+1}(\mu) = -P^T - (A_\mu^{(n)})^T \hat{u}_{\mathrm{du}}^{n+1}(\mu), \tag{4.21}$$

respectively. Define an auxiliary vector

$$\tilde{r}_{\mathrm{pr}}^{n+1} := b(u^n(\mu);\mu) - A_\mu^{(n)} \hat{u}^{n+1}(\mu) = A_\mu^{(n)} u^{n+1}(\mu) - A_\mu^{(n)} \hat{u}^{n+1}(\mu). \tag{4.22}$$

Note that the only difference of $\tilde{r}_{\mathrm{pr}}^{n+1}$ from r_{pr}^{n+1} is that $b(\hat{u}^n(\mu);\mu)$ in (4.20) is replaced by $b(u^n(\mu);\mu)$ in (4.22), so that we have a direct relation between $\tilde{r}_{\mathrm{pr}}^{n+1}$ and $u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)$,

the error of the approximate solution. This relation will aid the derivation of the error bound in Theorem 4.3.1. For simplicity, we define

$$\Phi_\mu^{n+1} := \|(A_\mu^{(n)})^{-T}\| \|r_{\mathrm{du}}^{n+1}\| + \|\hat{u}_{\mathrm{du}}^{n+1}(\mu)\|. \tag{4.23}$$

Theorem 4.3.1. *For the systems (4.16) and (4.18), assume that $A_\mu^{(n)}$ is invertible for any $\mu \in \mathcal{P}$. Then the output error $e_{\mathrm{O}}^{n+1}(\mu) = y^{n+1}(\mu) - \hat{y}^{n+1}(\mu)$ at the time instance t^{n+1} satisfies*

$$\|e_{\mathrm{O}}^{n+1}(\mu)\| \leq \Phi_\mu^{n+1} \|\tilde{r}_{\mathrm{pr}}^{n+1}\|, \quad n = 0, \ldots, K-1. \tag{4.24}$$

Proof. Left-multiplying both sides of (4.17) by $\left(u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)\right)^T$, we have

$$\left(u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)\right)^T (A_\mu^{(n)})^T u_{\mathrm{du}}^{n+1}(\mu) = -\left(u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)\right)^T P^T.$$

Transposing this equation, we obtain

$$\left(u_{\mathrm{du}}^{n+1}(\mu)\right)^T A_\mu^{(n)} \left(u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)\right) = -P\left(u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)\right). \tag{4.25}$$

By the definition of $\tilde{r}_{\mathrm{pr}}^{n+1}$, we have

$$\tilde{r}_{\mathrm{pr}}^{n+1} = A_\mu^{(n)} \left(u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)\right). \tag{4.26}$$

Left-multiplying both sides of (4.26) by $\left(u_{\mathrm{du}}^{n+1}(\mu)\right)^T$ yields

$$\left(u_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1} = \left(u_{\mathrm{du}}^{n+1}(\mu)\right)^T A_\mu^{(n)} \left(u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)\right). \tag{4.27}$$

Combining (4.25) and (4.27), we obtain

$$-P\left(u^{n+1}(\mu) - \hat{u}^{n+1}(\mu)\right) = \left(u_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1}.$$

Introducing a vector $\tilde{y}^{n+1}(\mu) = P\hat{u}^{n+1}(\mu) - \left(\hat{u}_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1}$, we have

$$\begin{aligned} |y^{n+1}(\mu) - \tilde{y}^{n+1}(\mu)| &= |Pu^{n+1}(\mu) - P\hat{u}^{n+1}(\mu) + \left(\hat{u}_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1}| \\ &= |-\left(u_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1} + \left(\hat{u}_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1}| \\ &= |-\left(u_{\mathrm{du}}^{n+1}(\mu) - \hat{u}_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1}| \\ &\leq \|u_{\mathrm{du}}^{n+1}(\mu) - \hat{u}_{\mathrm{du}}^{n+1}(\mu)\| \|\tilde{r}_{\mathrm{pr}}^{n+1}\|. \end{aligned} \tag{4.28}$$

By the definition of the residual in (4.21) and the dual system in (4.17), we have

$$\begin{aligned} r_{\mathrm{du}}^{n+1} &= -P^T - (A_\mu^{(n)})^T \hat{u}_{\mathrm{du}}^{n+1}(\mu) \\ &= (A_\mu^{(n)})^T u_{\mathrm{du}}^{n+1}(\mu) - (A_\mu^{(n)})^T \hat{u}_{\mathrm{du}}^{n+1}(\mu) \\ &= (A_\mu^{(n)})^T \left(u_{\mathrm{du}}^{n+1}(\mu) - \hat{u}_{\mathrm{du}}^{n+1}(\mu)\right). \end{aligned}$$

Since $A_\mu^{(n)}$ is invertible, we have

$$u_{\mathrm{du}}^{n+1}(\mu) - \hat{u}_{\mathrm{du}}^{n+1}(\mu) = (A_\mu^{(n)})^{-T} r_{\mathrm{du}}^{n+1}. \tag{4.29}$$

Combining (4.28) and (4.29), we obtain

$$|y^{n+1}(\mu) - \tilde{y}^{n+1}(\mu)| \leq \|(A_\mu^{(n)})^{-T} r_{\mathrm{du}}^{n+1}\| \|\tilde{r}_{\mathrm{pr}}^{n+1}\| \leq \|(A_\mu^{(n)})^{-T}\| \|r_{\mathrm{du}}^{n+1}\| \|\tilde{r}_{\mathrm{pr}}^{n+1}\|.$$

Thus

$$\begin{aligned} |y^{n+1}(\mu) - \hat{y}^{n+1}(\mu)| &= |y^{n+1}(\mu) - \tilde{y}^{n+1}(\mu) - \left(\hat{u}_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1}| \\ &\leq |y^{n+1}(\mu) - \tilde{y}^{n+1}(\mu)| + |\left(\hat{u}_{\mathrm{du}}^{n+1}(\mu)\right)^T \tilde{r}_{\mathrm{pr}}^{n+1}| \\ &\leq \|(A_\mu^{(n)})^{-T}\| \|r_{\mathrm{du}}^{n+1}\| \|\tilde{r}_{\mathrm{pr}}^{n+1}\| + \|\left(\hat{u}_{\mathrm{du}}^{n+1}(\mu)\right)^T\| \|\tilde{r}_{\mathrm{pr}}^{n+1}\| \\ &= \Phi_\mu^{n+1} \|\tilde{r}_{\mathrm{pr}}^{n+1}\|. \end{aligned} \tag{4.30}$$

□

Note that the error bound in (4.24) is not feasible to compute in practice, because the detailed solution $u^{n+1}(\mu)$ is involved in the evaluation of $\|\tilde{r}_{\mathrm{pr}}^{n+1}\|$. For this, defining

$$\rho^{n+1}(\mu) := \frac{\|\tilde{r}_{\mathrm{pr}}^{n+1}\|}{\|r_{\mathrm{pr}}^{n+1}\|}, \tag{4.31}$$

we have the following two corollaries, showing the existence of $\rho^{n+1}(\mu)$ by an upper bound and a lower bound, under certain assumptions. Consequently, the output error bound in Theorem 4.3.1 becomes

$$\|e_{\mathrm{O}}^{n+1}(\mu)\| \leq \Delta_N^{n+1}(\mu) := \Phi_\mu^{n+1} \rho^{n+1}(\mu) \|r_{\mathrm{pr}}^{n+1}\|. \tag{4.32}$$

Corollary 4.3.2. *Under the assumptions in Theorem 4.3.1, for the vectors $\{\tilde{r}_{\mathrm{pr}}^n\}_{n=1}^K$, assume that there exists a positive constant α such that*

$$\alpha \leq \frac{\|\tilde{r}_{\mathrm{pr}}^{n+1}\|}{\|\tilde{r}_{\mathrm{pr}}^n\|}, \quad n = 1, \ldots, K-1, \quad \mu \in \mathcal{P}. \tag{4.33}$$

Assume that for all $\mu \in \mathcal{P}$ the operator $b(\cdot;\mu)$ in (4.16) is Lipschitz continuous w.r.t. the first argument, i.e., there exists a positive constant L_b such that

$$\|b(u_1;\mu) - b(u_2;\mu)\| \leq L_b \|u_1 - u_2\|, \quad u_1, u_2 \in \mathcal{W}^{\mathcal{N}}, \quad \mu \in \mathcal{P}. \tag{4.34}$$

Assume further

$$L_b < \alpha / \|(A_\mu^{(n)})^{-1}\|, \quad n = 0, \ldots, K-1, \quad \mu \in \mathcal{P}. \tag{4.35}$$

Then

$$\underline{\rho}^{n+1}(\mu) \le \rho^{n+1}(\mu) \le \bar{\rho}^{n+1}(\mu), \tag{4.36}$$

where $\underline{\rho}^{n+1}(\mu) = \frac{\alpha}{\alpha + L_b\|(A_\mu^{(n-1)})^{-1}\|}$, $\bar{\rho}^{n+1}(\mu) = \frac{\alpha}{\alpha - L_b\|(A_\mu^{(n-1)})^{-1}\|}$, $n = 1, \ldots, K-1$, *and* $\mu \in \mathcal{P}$.

Proof. By the definition of the vectors r_{pr}^{n+1} and $\tilde{r}_{\mathrm{pr}}^{n+1}$ (in (4.20) and (4.22), respectively) and using the Lipschitz condition in (4.34), we have

$$\begin{aligned} \|r_{\mathrm{pr}}^{n+1} - \tilde{r}_{\mathrm{pr}}^{n+1}\| &= \|b(\hat{u}^n(\mu);\mu) - b(u^n(\mu);\mu)\| \\ &\le L_b\|\hat{u}^n(\mu) - u^n(\mu)\| = L_b\|(A_\mu^{(n-1)})^{-1}\tilde{r}_{\mathrm{pr}}^n\| \\ &\le L_b\|(A_\mu^{(n-1)})^{-1}\|\|\tilde{r}_{\mathrm{pr}}^n\|. \end{aligned} \tag{4.37}$$

By the inequality in (4.33), we have

$$\|\tilde{r}_{\mathrm{pr}}^n\| \le \|\tilde{r}_{\mathrm{pr}}^{n+1}\|/\alpha. \tag{4.38}$$

Substituting (4.38) into (4.37) and using the triangle inequality, we have

$$\|\tilde{r}_{\mathrm{pr}}^{n+1}\| - \|r_{\mathrm{pr}}^{n+1}\| \le \|r_{\mathrm{pr}}^{n+1} - \tilde{r}_{\mathrm{pr}}^{n+1}\| \le L_b\|(A_\mu^{(n-1)})^{-1}\|\|\tilde{r}_{\mathrm{pr}}^{n+1}\|/\alpha. \tag{4.39}$$

With simple calculations, we have

$$\frac{\|\tilde{r}_{\mathrm{pr}}^{n+1}\|}{\|r_{\mathrm{pr}}^{n+1}\|} \le \frac{\alpha}{\alpha - L_b\|(A_\mu^{(n-1)})^{-1}\|}, \tag{4.40}$$

i.e., the second inequality in (4.36) is thus proved. Analogously, replacing the left-hand side in (4.39) with $\|r_{\mathrm{pr}}^{n+1}\| - \|\tilde{r}_{\mathrm{pr}}^{n+1}\|$, yields the first inequality in (4.36). □

The assumption for L_b in (4.35) in Corollary 4.3.2 is reasonable only when $\|(A_\mu^{(n)})^{-1}\|$ is relatively small or moderate at most. When $\|(A_\mu^{(n)})^{-1}\|$ is large, we have the following corollary, where another upper and lower bounds for $\rho^n(\mu)$ is provided.

Corollary 4.3.3. *Under the assumptions in Theorem 4.3.1, for the vectors* $\{\tilde{r}_{\mathrm{pr}}^n\}_{n=1}^K$, *assume that there exist two positive constants* $\underline{\alpha}$, $\bar{\alpha}$ *such that*

$$\underline{\alpha} \le \frac{\|\tilde{r}_{\mathrm{pr}}^n\|}{\|\tilde{r}_{\mathrm{pr}}^{n+1}\|} \le \bar{\alpha}, \quad n = 1, \ldots, K-1, \quad \mu \in \mathcal{P}. \tag{4.41}$$

Assume that for all $\mu \in \mathcal{P}$ the operator $b(\cdot;\mu)$ in (4.16) is bi-Lipschitz continuous w.r.t. the first argument, i.e., there exist two positive constants $\underline{L}_b$, $\bar{L}_b$ such that

$$\underline{L}_b\|u_1 - u_2\| \le \|b(u_1;\mu) - b(u_2;\mu)\| \le \bar{L}_b\|u_1 - u_2\|, \quad u_1, u_2 \in \mathcal{W}^{\mathcal{N}}, \quad \mu \in \mathcal{P}. \tag{4.42}$$

Assume further

$$\underline{L}_b > \underline{\alpha}^{-1}/\|(A_\mu^{(n)})^{-1}\|, \quad n = 0, \ldots, K-1, \quad \mu \in \mathcal{P}. \tag{4.43}$$

Then

$$\underline{\rho}^{n+1}(\mu) \le \rho^{n+1}(\mu) \le \bar{\rho}^{n+1}(\mu), \tag{4.44}$$

where $\underline{\rho}^{n+1}(\mu) = \frac{1}{\bar{\alpha}\bar{L}_b\|(A_\mu^{(n-1)})^{-1}\|+1}$, $\bar{\rho}^{n+1}(\mu) = \frac{1}{\underline{\alpha}\,\underline{L}_b\|(A_\mu^{(n-1)})^{-1}\|-1}$, $n = 1, \ldots, K-1$, *and* $\mu \in \mathcal{P}$.

Proof. By the definition of the vectors r_{pr}^{n+1} and $\tilde{r}_{\mathrm{pr}}^{n+1}$ (in (4.20) and (4.22), respectively) and using the Lipschitz condition in (4.42), we have

$$\begin{aligned} \|r_{\mathrm{pr}}^{n+1} - \tilde{r}_{\mathrm{pr}}^{n+1}\| &= \|b(\hat{u}^n(\mu);\mu) - b(u^n(\mu);\mu)\| \\ &\ge \underline{L}_b\|\hat{u}^n(\mu) - u^n(\mu)\| = \underline{L}_b\|(A_\mu^{(n-1)})^{-1}\tilde{r}_{\mathrm{pr}}^n\| \\ &\ge \underline{L}_b\|(A_\mu^{(n-1)})^{-1}\|\|\tilde{r}_{\mathrm{pr}}^n\|. \end{aligned} \tag{4.45}$$

By the first inequality in (4.41), we have

$$\|\tilde{r}_{\mathrm{pr}}^n\| \ge \underline{\alpha}\|\tilde{r}_{\mathrm{pr}}^{n+1}\|. \tag{4.46}$$

Substituting (4.46) into (4.45) and using the triangle inequality, we have

$$\|\tilde{r}_{\mathrm{pr}}^{n+1}\| + \|r_{\mathrm{pr}}^{n+1}\| \ge \|r_{\mathrm{pr}}^{n+1} - \tilde{r}_{\mathrm{pr}}^{n+1}\| \ge \underline{\alpha}\,\underline{L}_b\|(A_\mu^{(n-1)})^{-1}\|\|\tilde{r}_{\mathrm{pr}}^{n+1}\|,$$

so

$$(\underline{\alpha}\,\underline{L}_b\|(A_\mu^{(n-1)})^{-1}\| - 1)\|\tilde{r}_{\mathrm{pr}}^{n+1}\| \le \|r_{\mathrm{pr}}^{n+1}\|,$$

which implies that the second inequality in (4.44) holds. For the first inequality in (4.44), analogous to (4.39), we have

$$\|r_{\mathrm{pr}}^{n+1}\| - \|\tilde{r}_{\mathrm{pr}}^{n+1}\| \le \|r_{\mathrm{pr}}^{n+1} - \tilde{r}_{\mathrm{pr}}^{n+1}\| \le \bar{\alpha}\bar{L}_b\|(A_\mu^{(n-1)})^{-1}\|\|\tilde{r}_{\mathrm{pr}}^{n+1}\|,$$

due to the new assumptions in (4.41) and (4.42). The first inequality in (4.44) is thus proved. □

Although the bi-Lipschitz continuity is required in Corollary 4.3.3, no restriction is imposed on the upper Lipschitz constant $\bar{L}_b$. Moreover, the restriction on $\underline{L}_b$ in (4.42) is actually not strong if $\|(A_\mu^{(n)})^{-1}\|$ is large. For example, for systems that are nearly noncoercive, i.e., when $A_\mu^{(n)}$ is close to singular, $\|(A_\mu^{(n)})^{-1}\|$ can be of $\mathcal{O}(10^3)$, or even larger.

Note that (4.36) and (4.44) hold for $n = 1, \ldots, K-1$. When $n = 0$, $\|\tilde{r}_{\rm pr}^1\|/\|r_{\rm pr}^1\| = 1$ for zero initial conditions, i.e., $u(0;\mu) \equiv 0$. For problems with nonzero initial conditions, i.e., $u(0;\mu) \neq 0$, one can use the variable transformation $\tilde{u}(\mu) = u(\mu) - u(0;\mu)$ to derive a transformed system with zero initial conditions; then the same conclusion can be similarly obtained for the transfered system.

Remark 4.3.4. *From Corollaries 4.3.2 and 4.3.3, we know that the quantity $\rho^{n+1}(\mu)$ in* (4.32) *is bounded. One may use an upper bound $\bar{\rho}^{n+1}(\mu)$ in* (4.36) *or* (4.44) *to derive an output error bound, namely,*

$$\|e_{\rm O}^{n+1}(\mu)\| \leq \Phi_\mu^{n+1} \bar{\rho}^{n+1}(\mu) \|r_{\rm pr}^{n+1}\|. \tag{4.47}$$

However, computing $\bar{\rho}^{n+1}(\mu)$ involves computing the Lipschitz constant L_b (or $\underline{L}_b$) and α (or $\underline{\alpha}$), which are nevertheless not practically computable.

Alternatively, for an efficient computation, one can directly estimate $\rho^{n+1}(\mu)$ by observing the maximal ratio among all the time steps $\max\limits_{k\in\{1,\ldots,K\}}\{\rho^k(\mu_\star)\}$, or the average ratio $\frac{1}{K}\sum_{k=1}^K \rho^k(\mu_\star)$ when the average of the output errors is estimated. Here, $\mu_\star$ is the parameter selected by the greedy algorithm, to be addressed in the following section. To compute the quantity $\tilde{r}_{\rm pr}^k(\mu_\star)$ for $\rho^k(\mu_\star)$, the detailed solutions $u^k(\mu_\star)$, $k = 1, \ldots, K$, at $\mu_\star$ are required, which cause no additional cost for snapshot-based MOR methods because the detailed solutions at this parameter $\mu_\star$ are already available after the RB extension. Although the parameter $\mu_\star$, which causes the largest error (measured by the error estimation) in the parameter domain, may not be the one that causes the largest ratio $\rho^{n+1}(\mu)$, it makes sense to use the data at $\mu_\star$ to estimate $\rho^{n+1}(\mu)$. It should be pointed out that such an estimation on $\rho^{n+1}(\mu)$ can result in a sharp estimate but may sacrifice the rigorousness. Thus, we may say that the error "bound" obtained by estimating $\rho^{n+1}(\mu)$ is only an output error estimation.

Remark 4.3.5. *When the operator $b(\cdot;\mu)$ is nonlinear w.r.t. the first argument and/or nonaffine w.r.t. the parameter μ, the empirical interpolation [17] can be employed. The ROM can be formulated following* (3.27)*. In such a case, the term $\|r_{\rm pr}^{n+1}\|$ in (4.32) can*

be further bounded using the EI error bound, i.e.,

$$\begin{aligned}\|r_{\mathrm{pr}}^{n+1}\| &= \|B_\mu^{(n)}\hat{u}^n(\mu) + g(\hat{u}^n(\mu);\mu) - A_\mu^{(n)}\hat{u}^{n+1}(\mu)\| \\ &= \|B_\mu^{(n)}\hat{u}^n(\mu) + \mathcal{I}_M[g(\hat{u}^n(\mu);\mu)] - A_\mu^{(n)}\hat{u}^{n+1}(\mu) \\ &\qquad + g(\hat{u}^n(\mu);\mu) - \mathcal{I}_M[g(\hat{u}^n(\mu);\mu)]\| \\ &\le \|r^{n+1}(\mu)\| + \|g(\hat{u}^n(\mu);\mu) - \mathcal{I}_M[g(\hat{u}^n(\mu);\mu)]\| \\ &\le \|r^{n+1}(\mu)\| + \epsilon_{\mathrm{EI}}^n(\mu),\end{aligned}$$

where $r^{n+1}(\mu)$ is defined in (4.2), and $\epsilon_{\mathrm{EI}}^n(\mu)$ is the error due to the EI, as defined in (4.5).

To compute the error bound in (4.32), one needs to efficiently compute the norm of the matrix inverse $(A_\mu^{(n)})^{-T}$. The following remark addresses how to evaluate $\|(A_\mu^{(n)})^{-T}\|$.

Remark 4.3.6. *If the vector norm is taken as the standard 2-norm, e.g., when the discrete system is obtained by the finite volume or finite difference discretization, the matrix norm $\|(A_\mu^{(n)})^{-T}\|$ is the spectral norm of $(A_\mu^{(n)})^{-T}$. Therefore,*

$$\|(A_\mu^{(n)})^{-T}\|_2 = \|(A_\mu^{(n)})^{-1}\|_2 = \sigma_{\max}\left((A_\mu^{(n)})^{-1}\right) = \frac{1}{\sigma_{\min}(A_\mu^{(n)})}, \tag{4.48}$$

the reciprocal of the smallest singular value of $A_\mu^{(n)}$. For some special cases in which the matrix $A_\mu^{(n)}$ is a constant matrix, the smallest singular value of A is computed once and can be used repeatedly.

For the general vector norm $\|\cdot\|_{\mathsf{H}}$, induced by the inner product $\langle v_1, v_2\rangle := v_1^T\mathsf{H}v_2$, $v_1, v_2 \in \mathcal{W}^{\mathcal{N}}$, where H is a symmetric positive definite matrix, e.g., the mass matrix in the finite element discretization, the induced matrix norm can be defined as

$$\|Z\|_{\mathsf{H}} := \max_{\|x\|=1}\|Zx\|_{\mathsf{H}} = \max_{\|x\|=1}\sqrt{x^TZ^T\mathsf{H}Zx} = \|Z^T\mathsf{H}Z\|_2, \quad Z \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}.$$

This implies that

$$\begin{aligned}\|Z^T\mathsf{H}Z\|_2 &= \sqrt{\lambda_{\max}\left(\left(Z^T\mathsf{H}Z\right)^T Z^T\mathsf{H}Z\right)} \\ &= \lambda_{\max}(Z^T\mathsf{H}Z) \\ &= \lambda_{\max}\left(Z^T\mathsf{L}^T\mathsf{L}Z\right) \\ &= \sigma_{\max}^2(\mathsf{L}Z).\end{aligned}$$

Here L *is a lower triangular matrix of the Cholesky factorization of* H*, i.e.* $\mathsf{L}^T\mathsf{L} = \mathsf{H}$*;* $\lambda_{\max}(\cdot)$ *refers to the largest eigenvalue of a matrix. Thus,* $\|(A_\mu^{(n)})^{-T}\|_{\mathsf{H}}$ *can be obtained by*

$$\|(A_\mu^{(n)})^{-T}\|_{\mathsf{H}} = \sigma_{\max}^2\left(\mathsf{L}(A_\mu^{(n)})^{-T}\right) = \frac{1}{\sigma_{\min}^2((A_\mu^{(n)})^T\mathsf{L}^{-1})}.$$

Remark 4.3.7. *The assumptions on the Lipschitz constants (i.e.,* L_b *and* $\underline{L}_b$*) in Corollary 4.3.2 and 4.3.3 require that* α *and* $\underline{\alpha}$ *cannot be too small. This can be achieved if the time step of the detailed simulation is well chosen. In fact, a well-chosen time step results in an even distribution of the error of the solution to the FOM over the time interval, and this property can be inherited by the solution to the ROM [95]. In addition, the values of* α*,* $\underline{\alpha}$*, and* $\bar{\alpha}$ *for the test examples are* $\mathcal{O}(1)$*, as will be shown in numerical results in the next section.*

Remark 4.3.8. *For the case of multiple outputs, i.e.,* $N_{\mathrm{O}} > 1$*, an error bound for each component of the output vector can be obtained from Theorem 4.3.1. The final error bound for the whole vector of outputs can be taken as the maximum of all the error bounds.*

Note that the error bound is independent of the projection matrix pairs $(V_{\mathrm{pr}}, W_{\mathrm{pr}})$ and $(V_{\mathrm{du}}, W_{\mathrm{du}})$. It is applicable to any projection-based MOR method. In addition, if one takes $W_{\mathrm{pr}} = V_{\mathrm{pr}}$, then the ROM can be obtained by using Galerkin projection, as is usually implemented by the RBM.

4.4 Performance of the output error bound/estimation

To show the performance of the error estimators presented earlier, we consider two academic examples. One is a linear convection-diffusion equation in [69], which is used to compare the performance of the new error estimation and the error bound using the existing primal-dual approach [69, 72]. The other is the viscous Burgers' equation, which is used to demonstrate that our method is applicable to a large class of nonlinear evolution equations. More numerical results for applications of the error estimation to real-life models in chemical engineering will be given in Chapter 6.

The RBM presented in Chapter 3 is employed to construct the ROMs for all the examples. More specifically, Algorithm 2 is used to generate the RB matrix V, and the ROMs are

constructed by the Galerkin projection. For comparison, we use the proposed output error estimation and other existing output error bounds to define the error indicators, and to construct the ROMs, respectively. The error indicator is defined as $\psi_N(\mu) := \frac{1}{K}\sum_{n=1}^{K}\Psi_N^n(\mu)$, where $\Psi_N^n(\mu)$ is the corresponding output error bound/estimation for the parameter μ at the time instance t^n. For example, when the primal-dual output error estimation is employed, $\Psi_N^n(\mu) = \Delta_N^n(\mu)$, where $\Delta_N^n(\mu)$ is defined in (4.32). The error indicator $\psi_N(\mu)$ is used to measure the average output error (i.e., $\frac{1}{K}\sum_{n=1}^{K}\|y(u^n(\mu)) - y(\hat{u}^n(\mu))\|$) over the whole evolution process.

In what follows, PO-EB refers to the primal-only error bound in (4.12), PD-ES refers to the primal-dual output error estimation in (4.32), and PD-EB refers to the primal-dual error bound in [72]. To compute the new output error estimation, the quantity $\rho^{n+1}(\mu)$ in (4.32) needs to be estimated, as discussed in Remark 4.3.4. In this work, we use the average ratio $\tilde{\rho}_N^\star := \frac{1}{K}\sum_{k=1}^{K}\rho^k(\mu_\star)$ to estimate $\rho^{n+1}(\mu)$, since we measure the average of the output errors over time. After each iteration of the greedy algorithm, we compute the average ratio $\tilde{\rho}_N^\star$ at the selected parameter $\mu_\star$ and use it as an estimate of $\rho^{n+1}(\mu)$ for the next iteration. All the computations were carried out using C++ code on a PC with an Intel Core 2 Quad CPU Q9550 2.83 GHz 4.00 GB RAM.

4.4.1 Linear convection-diffusion equation

In this section, we consider a linear convection-diffusion equation which models the movement of fluids and other transport phenomena. This model is used as a test case for the primal-dual error bound in [69]. Here, we use it to compare the performance of the proposed primal-dual error estimation (PD-ES) in (4.32) and the existing primal-dual error bound (PD-EB) in [72].

Model description and reduced-order modeling

The governing equation for this model is given as

$$u_t = q_1 u_{xx} + q_2 u_x - q_2, \quad x \in \Omega := (0,1), \quad t \in (0,T].$$

The initial and boundary conditions are specified as follows:

$$u(0,x) = -2x^2 + 2x; \quad u(t,0) = u(t,1) = 0, \quad t > 0.$$

The output of interest is the average value of u over a small interval Ω_0 as a function of time, i.e., $y(u(t)) := \frac{1}{|\Omega_0|}\int_{\Omega_0} u(t,x)\,dx$, $\Omega_0 := [0.495, 0.505]$.

In this model, we choose the diffusivity q_1 and the velocity q_2 as the parameters, i.e., $\mu := (q_1, q_2)$. The interesting parameter domain is chosen as $\mathcal{P} := [0.1, 1] \times [0.5, 5]$. To construct the FOM, we use the finite volume method for the spatial discretization and the backward Euler scheme for the temporal discretization. We choose $\mathcal{N} = 800$ as the number of degrees of freedom for the FOM and an equal time step $\Delta t = T/K$, $T = 1, K = 100$. The FOM is of the general form in (3.22), except that no nonlinear term is involved. Algorithm 2 is employed to construct the ROMs with the two aforementioned error estimators: PD-EB and PD-ES.

Results

The training set $\mathcal{P}_{\text{train}}$ consists of 200 sample points randomly distributed in the parameter domain $\mathcal{P}$. Figure 4.1 shows the decay of both error estimations and the corresponding true error for the output during the RB construction process. ROM-1 and ROM-2 are the ROMs constructed by using PD-EB and PD-ES, respectively. It is seen that the proposed PD-ES outperforms the existing PD-EB.

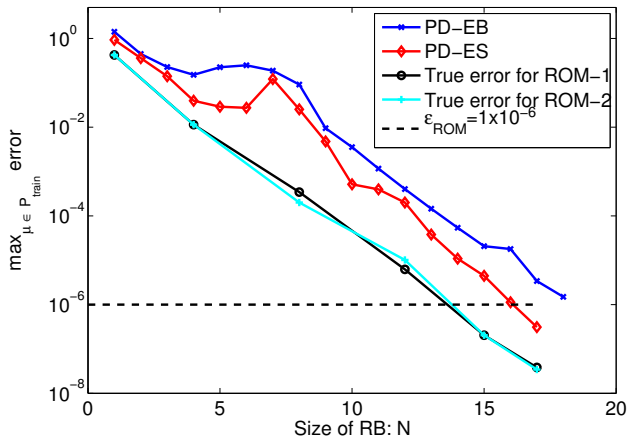


Figure 4.1: Decay of the existing primal-dual error bound (PD-EB), the proposed primal-dual error estimation (PD-ES), and the corresponding true error during the RB construction process for the linear convection-diffusion equation.

As mentioned above, the estimation of $\rho^n(\mu)$ (by $\tilde{\rho}_N^\star$) is based on the fact that it is bounded, as shown in Corollaries 4.3.2 and 4.3.3. We have carefully checked the assumptions made in the two corollaries, and the results are detailed as follows. First, we found that all values of $\|(A_\mu^{(n)})^{-1}\|$ are in the interval $[0.9, 1.0]$. Second, we plot the ratio $\|\tilde{r}_{\text{pr}}^{n+1}\|/\|\tilde{r}_{\text{pr}}^{n}\|$ at the chosen parameter $\mu_\star$ as a function of the time index t^n for three different RB dimensions in Figure 4.2. It can be seen that all the values of the ratio are in the range of $[0.4, 3]$. In fact, the ratio at other RB dimensions is pretty similar, i.e., it is always in the range in $\mathcal{O}(1)$. This means that the constant α in (4.33) exists, and it is also in $\mathcal{O}(1)$. As a result, $\alpha/\|(A_\mu^{(n)})^{-1}\| \approx \mathcal{O}(1)$, and the condition on the Lipschitz constant in (4.35) becomes $L_b \lesssim 1$, which is reasonable for a linear continuous operator, as here for this example $L_b = 1$. Thus, all the assumptions in Corollary 4.3.2 are satisfied, so the quantity $\rho^n(\mu)$ is bounded, and in turn, using $\tilde{\rho}_N^\star$ as an estimate of $\rho^n(\mu)$ is practical and meaningful. Figure 4.3 shows the average ratio $\tilde{\rho}_N^\star$ as a function of the RB size N. We see that it converges to 1 as the RB is extended.

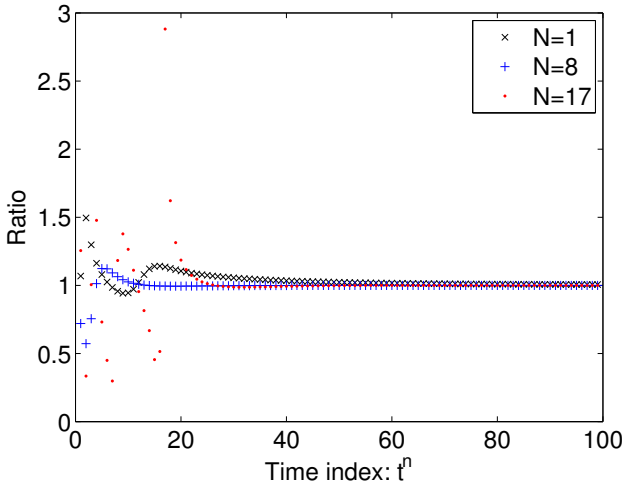


Figure 4.2: Behavior of the ratio $\frac{\|\tilde{r}^{n+1}\|}{\|\tilde{r}^n\|}$ in the time trajectory corresponding to different RB dimensions for the linear convection-diffusion equation.

4.4.2 Burgers' equation

The Burgers' equation describes the fundamental nonlinear phenomena in fluid dynamics and is often considered as the starting point to test a new algorithm for nonlinear problems.

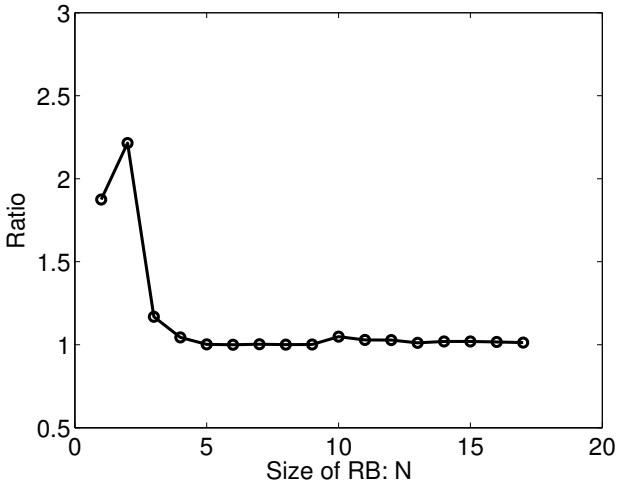


Figure 4.3: Behavior of the average ratio $\tilde{\rho}_N^\star$ during the RB construction process for the linear convection-diffusion equation.

We now use the unsteady viscous Burgers' equation to show that the proposed error estimation is applicable for MOR of general nonlinear evolution equations. We further compare the proposed primal-dual error estimation (PD-ES) with the proposed primal-only error bound (PO-EB).

Reduced-order modeling of Burgers' equation

In this work, we consider the unsteady viscous Burgers' equation as follows:

$$u_t + \left(\frac{u^2}{2}\right)_x = \nu u_{xx} + s(u,x), \quad x \in (0,1), \quad t \in (0,T], \tag{4.49}$$

where $\nu \in \mathcal{P}$ is the viscosity coefficient, and $s(u,x)$ is the source term. The output of interest is the value of u at $x = 1$ as a function of t, i.e., $y(t;\nu) := u(t,1;\nu)$.

In this model, the viscosity coefficient ν is considered as the parameter, i.e., $\mu := \nu$. Note that the computation becomes more challenging when ν is smaller, e.g., $\nu \approx \mathcal{O}(10^{-3})$, because the instability grows exponentially with the evolution time [132]. For MOR, it becomes more challenging when a smaller value of ν is involved. To numerically verify this, we choose two parameter domains: $\tilde{\mathcal{P}} = [0.05, 1]$ and $\mathcal{P} = [0.001, 1]$. We will see that the ROM has a better convergence rate and other good properties when $\tilde{\mathcal{P}}$ with a larger

value of ν is employed. We take $T = 2$ and $s(u, x) \equiv 1$ in the following computations. For discretization, we use the finite volume method to construct the FOM, in the general form of (3.22).

An *a posteriori* error estimation for the RBM applied to this equation is proposed in [132], where the successive constraint method was used to estimate the lower bound of the stability constant. The error estimation is actually a summation over time of the dual norm of the residual. As pointed out in [132], this error estimation is no longer useful, when the viscosity ν is small and the final time T is large. In addition, this error estimation is applicable to problems that are at most quadratically nonlinear. By contrast, the newly proposed error estimation is applicable to MOR of general nonlinear evolution equations.

Results

The following results are obtained by using the following initial and boundary conditions:

$$u(0, x) = 0, \quad x \in [0, 1]; \quad u(t, 0) = 0, \quad u_x(t, x)|_{x=1} = 0.$$

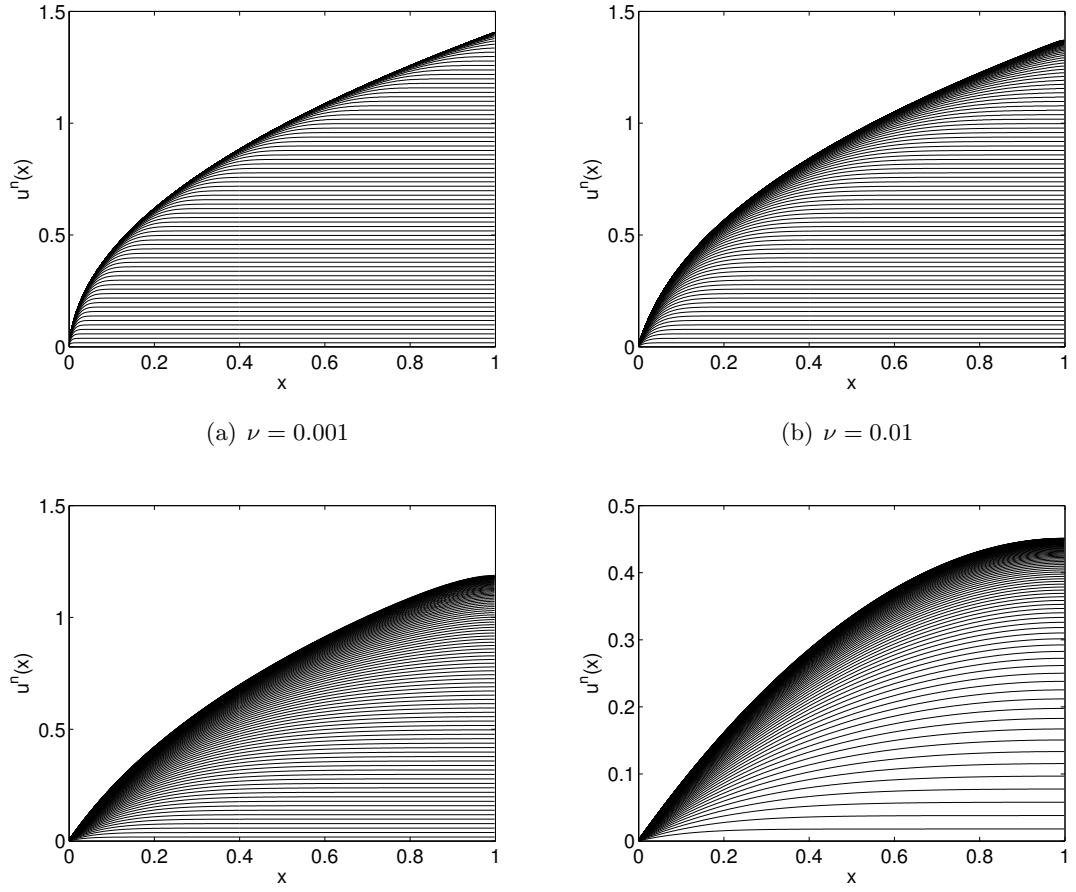
We use a uniform spatial grid with $\mathcal{N} = 500$ cells for the FOM, and $\Delta t = T/K, K = 1000$ for both the FOM and ROM simulations.

Figure 4.4 shows the solutions to the FOM as a function of x and t^n. Each line represents the solution $u(x, t^n)$ at the time instance $t = t^n$, $n = 10j$, $j = 0, \ldots, K/10$. The evolution process tends to be steady at final time. For the ROM construction, we choose a training set with 70 sample points log-uniformly distributed in the parameter domain $\mathcal{P}$, to build the RB and the basis for the EI, respectively.

The behavior of PO-EB, PD-ES, and the corresponding true error are illustrated in Figure 4.5. It is seen that in both cases PD-ES works much better than PO-EB. Moreover, PD-ES is fairly sharp in comparison with the true error. By comparing the two figures in Figure 4.5, we can see that the convergence rate becomes relatively slow and many more basis vectors are needed to achieve certain accuracy, when smaller viscosity coefficient ν is involved. Since this is just an academic numerical example, there is not much computational time reduction from the ROM. For runtime comparison, we will report the computational time for the more challenging problems in Chapter 6.

As discussed in Remark 4.3.4, the constant $\rho^n(\mu)$ can be estimated based on the observation of the average ratio over all the time steps at the selected parameter $\mu_\star$ at each iteration step of the greedy algorithm. Note that the ratio is changing with the dimension of the RB and with the parameter $\mu_\star$ selected at each iteration step. The behavior of the ratio during the RB extension process is illustrated in Figure 4.6. From Figure 4.6(a), we see that the ratio decreases ("almost monotonically") as the RB is extended, which demonstrates that the difference between $\|\tilde{r}_{\mathrm{pr}}^{n+1}\|$ and $\|r_{\mathrm{pr}}^{n+1}\|$ becomes small as the accuracy of the ROM is increased. However, when smaller viscosity coefficient ν is involved, the ratio oscillates during the basis extension process, as shown in Figure 4.6(b). This is probably because the instability grows too fast when ν is small. In most cases, the value of the ratio is of the magnitude $\mathcal{O}(1)$, when the accuracy of the ROM achieves a certain degree, which will be further justified in the following examples (see Figures 6.3 and 6.7).

As addressed in Remark 4.3.7, with well-chosen time steps, the approximation errors $(\hat{u}^n(\mu) - u^n(\mu))$ can be evenly distributed in the time trajectory so that the norm of the vectors $\tilde{r}_{\mathrm{pr}}^{n+1} = A_\mu^{(n)} u^{n+1}(\mu) - A_\mu^{(n)} \hat{u}^{n+1}(\mu)$ is of the same magnitude over time, i.e., the assumptions in (4.35) and (4.43) (in Corollaries 4.3.2 and 4.3.3, respectively) are fulfilled. To numerically verify this, we plot the ratio $\frac{\|\tilde{r}_{\mathrm{pr}}^{n+1}\|}{\|\tilde{r}_{\mathrm{pr}}^{n}\|}$ as a function of time instant t^n for different RB dimensions in Figure 4.7. It is seen that the ratio is in the range of $[0.85, 1.5]$, which implies that the constants α, $\underline{\alpha}$ and $\bar{\alpha}$ are all of the magnitude of $\mathcal{O}(1)$. In addition, based on our discretization scheme, the norm of the matrix inverse $\|(A_\mu^{(n)})^{-1}\|$ are all in range of $[0.95, 1]$ for all $\mu \in \mathcal{P}_{\mathrm{train}}$. Thus, $\alpha/\|(A_\mu^{(n)})^{-1}\| \approx \mathcal{O}(1)$, which means that the assumption on the Lipschitz constant L_b in (4.34) in Corollary 4.3.2 is reasonable.



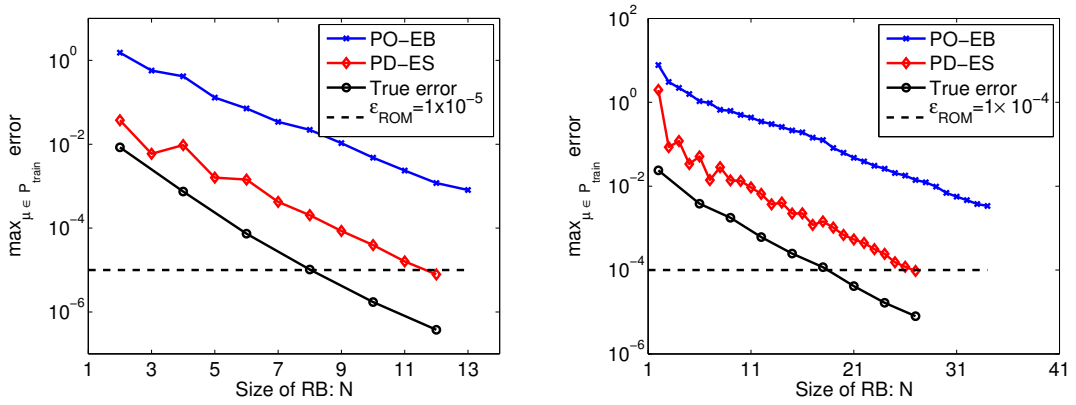
(a) $\nu = 0.001$

(b) $\nu = 0.01$

(c) $\nu = 0.1$

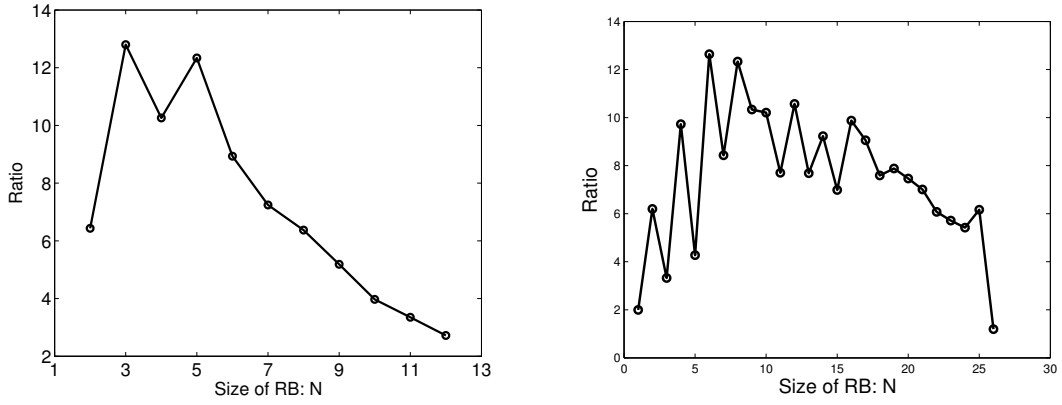
(d) $\nu = 1.0$

Figure 4.4: Solution to the Burgers' equation as a function of x and t^n with different viscosity coefficients ν. Each line represents the solution $u(x, t^n)$ at the time instance $t = t^n$, $n = 10j$, $j = 0, \ldots, K/10$.



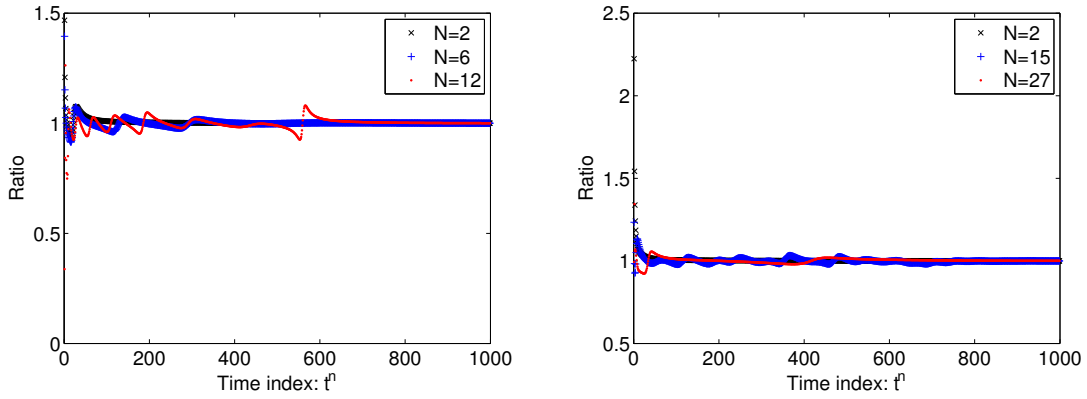
(a) $\nu \in \tilde{\mathcal{P}}$ (b) $\nu \in \mathcal{P}$

Figure 4.5: Decay of the primal-dual error bound (PO-EB), the primal-dual error estimation (PD-ES), and the corresponding true error during the RB construction process for the Burgers' equation. (a) $\nu \in \tilde{\mathcal{P}} = [0.05, 1]$; (b) $\nu \in \mathcal{P} = [0.001, 1]$.



(a) $\nu \in \tilde{\mathcal{P}}$ (b) $\nu \in \mathcal{P}$

Figure 4.6: Behavior of the average ratio $\tilde{\rho}_N^\star$ during the RB construction process for the Burgers' equation. (a) $\nu \in \tilde{\mathcal{P}} = [0.05, 1]$; (b) $\nu \in \mathcal{P} = [0.001, 1]$.



(a) $\nu \in \tilde{\mathcal{P}}$ (b) $\nu \in \mathcal{P}$

Figure 4.7: Behavior of the ratio $\frac{\|\tilde{r}^{n+1}\|}{\|\tilde{r}^n\|}$ in the time trajectory corresponding to different RB dimensions for the Burgers' equation. (a) $\nu \in \tilde{\mathcal{P}} = [0.05, 1]$; (b) $\nu \in \mathcal{P} = [0.001, 1]$.

Chapter 5

Acceleration and Adaptivity for RB Construction

Contents

(P)MOR aims at constructing a ROM that is used as a surrogate of the FOM to reproduce its main characteristics cheaply. Often, the construction of the ROM and the simulation of the ROM can be implemented in the strategy of offline-online decomposition, as discussed in Chapter 3. However, the offline computation can be fairly expensive, i.e., constructing such a ROM is often costly, especially for snapshot-based MOR methods because simulations based on the FOM need to be performed many times to collect the snapshots for constructing the RB.

In this chapter, we propose two strategies to reduce the offline cost. The first is a technique called ASS, which is used to collect the snapshots effectively so that the selected snapshots contain only the most representative information with a relatively small number of vectors. As a result, the basis can be efficiently constructed. The other is to accelerate FOM simulation using ROMs as predictors during the RB construction process, which is typically useful for multi-state systems. As discussed previously, the RB is often iteratively constructed for parametric systems. At each iteration, a FOM simulation has to be performed at a chosen parameter to enrich the current RB. Thus, reducing the runtime of

FOM simulation will lead to reduced computational time for the RB construction.

5.1 Motivation and previous work

ROMs are employed in many applications due to its computational advantages, such as small size (order) and the ability of reproducing the main dynamics of the original large-scale model. However, constructing such a ROM is often costly, especially for snapshot-based MOR methods. As shown in Algorithm 2, many simulations of the FOM need to be performed to collect the snapshots. Moreover, the quality of the snapshots has great influence on the quality of the ROM, as discussed in Section 3.3.2. If the collected snapshots do not contain sufficient system information, the resulting ROM will most likely to be of low accuracy. Therefore, effective collection of snapshots is crucial for snapshot-based MOR methods, and it determines the efficiency of the subsequent computation and the accuracy of the resulting ROM.

As mentioned earlier, the snapshots are usually collected from the solutions to the FOM (for RB generation) or function evaluations (for CRB generation) at different parameter samples in the training set and/or different time instances. For the generation of the RB or CRB, a training set $\mathcal{P}_{\text{train}}$ or $\mathcal{P}_{\text{train}}^{\text{crb}}$ of parameters is usually determined *a priori*. On the one hand, the size of the training set is desired to be large so that sufficient information of the parametric system can be collected. On the other hand, the RB or CRB should be generated efficiently.

To reduce the cost for RB generation, many efforts have been made in the last decade. These include, e.g., the *hp* certified RBM [53], adaptive grid partition in parameter space [82, 83], and the greedy-based adaptive sampling approach for MOR by using model-constrained optimization [34]. More recently, an adaptive greedy procedure is proposed for constructing the optimal training set in [141]. In these papers, the authors intend to choose the sample points adaptively and get an "optimal" training set. The "optimal" training set means that the original manifold $\mathcal{M} := \{u(\mu) \mid \mu \in \mathcal{P}\}$ can be well represented by the submanifold $\hat{\mathcal{M}} := \{u(\mu) \mid \mu \in \mathcal{P}_{\text{train}}\}$ induced by the sample set $\mathcal{P}_{\text{train}}$ with its size as small as possible.

In particular, for time-dependent problems, if the dynamics are of interest, the solution at

the time instances should be collected as snapshots. In such a case, even with the "optimal" training set, the number of snapshots can be huge if the total number of time steps for a single parameter is large. Such problems arise from, e.g., chemical engineering, fluid dynamics, and aerodynamics. A large number of snapshots lead to expensive subsequent computations. For example, for the generation of the RB, the POD mode in Step 4 of Algorithm 2 is hard to compute from the SVD of $\bar{U}$, due to the large size of the matrix $\bar{U}$. For the generation of the CRB, e.g., in Step 5 of Algorithm 3, the cost for solving the $M \times M$ systems increases drastically as the dimension M increases, because the number of the systems, i.e., the number of vectors in the set $\mathcal{G} = \{g(u^n(\mu); \mu) \mid \mu \in \mathcal{P}^{\text{crb}}_{\text{train}}, n = 0, \ldots, K\}$, is large. On the other hand, if we just trivially select parts of the solutions, e.g., at every two or several time steps, the resulting ROM might be of low accuracy because important information may have been lost due to such a naive snapshot selection. To circumvent this problem, we introduce an adaptive technique in the next section.

5.2 Adaptive snapshot selection

To further reduce the cost and complexity of the offline computation, we propose a technique of adaptive snapshot selection that we call ASS for the generation of the RB and/or the CRB. The basic idea of ASS was introduced in our recent work [25], and it is extended into a more general framework in [187].

For an "optimal" or a selected training set, we propose to select the snapshots adaptively according to the variation of the trajectory $S = \{v^n\}_{n=0}^K$, e.g., $\{u^n(\mu)\}_{n=0}^K$ or $\{g(u^n(\mu); \mu)\}_{n=0}^K$. The idea is to discard the redundant ("close to" linearly dependent) information from the trajectory. As is known that the snapshots are used to construct a (linearly independent) basis for MOR, therefore, the linearly dependent information in the snapshots should be discarded as early as possible. More precisely, assume that S^{A} is a subspace spanned by the selected snapshots, and v is a tested vector in the trajectory. It is unnecessary to include the vector v as a new snapshot if it is linearly dependent on S^{A}. To this end, define an indicator $\phi(S^{\text{A}}, v)$, which is used to measure the linear dependence of S^{A} and v. Then v is taken as a new snapshot only when v cannot be linearly represented by the vectors in S^{A}, by checking whether $\phi(S^{\text{A}}, v)$ satisfies certain conditions. Given a tolerance ε_{ASS}, assume that $\phi(S^{\text{A}}, v) \leq \varepsilon_{\text{ASS}}$ indicates they are "almost" linearly depen-

dent, i.e., v can be "almost" linearly represented by the vectors of S^{A}, the ASS process is realized with the following Algorithm 4.

Algorithm 4 Adaptive snapshot selection (ASS)

Input: $\{v^n\}_{n=0}^K$, $\varepsilon_{\mathrm{ASS}}$.

Output: Selected snapshot matrix $S^{\mathrm{A}} = [v^{n_1}, \ldots, v^{n_\ell}]$.

1: Initialization: $j = 1$, $n_j = 0$, $S^{\mathrm{A}} = [v^{n_j}]$.
2: **for** $n = 1, \ldots, K$ **do**
3: **if** $\phi\left(S^{\mathrm{A}}, v\right) > \varepsilon_{\mathrm{ASS}}$ **then**
4: $j = j + 1$.
5: $n_j = n$.
6: $S^{\mathrm{A}} = [S^{\mathrm{A}}, v^{n_j}]$.
7: **end if**
8: **end for**

In practice, the indicator can be defined as the angle between the vector v and the subspace spanned by the selected snapshots S^{A}, i.e., $\phi(S^{\mathrm{A}}, v) := \angle(S^{\mathrm{A}}, v)$. When $\phi(S^{\mathrm{A}}, v)$ is large, the correlation between S^{A} and v is weak. Additionally, one can use a relaxed but cheaper condition to measure the linear dependence, i.e., $\phi(S^{\mathrm{A}}, v) := \angle(v^{n_j}, v)$, where v^{n_j} is the last *selected* snapshot. This recovers the original ASS in [25].

Here we emphasize that v^{n_j} is the last *selected* snapshot rather than the last snapshot in the trajectory. Quite often, the "last selected" snapshot is not the last snapshot in the trajectory. For example, assume that at the 10th iteration step, two snapshots have been selected already, then at the 11th iteration step, we check the angle between the current tested vector v and the second selected snapshot, rather than the angle between v and the 10th snapshot (the last snapshot in the trajectory). In this way, the deleted vector v is always the one that is "almost" linearly dependent to the *selected* snapshot. Once it cannot be well represented by the last *selected* snapshot, it will always be selected by the ASS algorithm, though it might be almost linearly dependent to the 10th snapshot, i.e. the last snapshot in the trajectory.

Remark 5.2.1. *To make sure that any vector which cannot be fully represented by the selected snapshots is selected, and in turn, all the selected snapshots represent the information of the whole trajectory as complete as possible, we take a very small value of the tolerance parameter $\varepsilon_{\mathrm{ASS}}$ in the algorithm. However, the optimal value of the tolerance*

may be problem dependent and is thus more or less heuristic.

The ASS technique can be easily combined with the aforementioned algorithms for RB and CRB generation. For example, Algorithm 5 shows the combination of ASS with the POD-Greedy algorithm (Algorithm 2). In comparison with the original Algorithm 2, only one step is additional, while the number of the selected snapshots is largely reduced. When the ASS technique is employed for the construction of the CRB, much fewer vectors in $\mathcal{G} = \{g(u^n(\mu);\mu) \mid \mu \in \mathcal{P}^{\mathrm{crb}}_{\mathrm{train}}, n = 0, \ldots, K\}$ are collected as snapshots, so the subsequent computation is comparatively cheap. In Algorithm 5, the ASS technique serves as a pretreatment of the snapshots in the sense that it produces a much thinner matrix of selected snapshots, so that the matrix is suitable for SVD with the limited computer memory. It can be considered as preprocessing for SVD.

Algorithm 5 RB generation using ASS-POD-Greedy

Input: $\mathcal{P}_{\mathrm{train}}$, μ_0, $\varepsilon_{\mathrm{ROM}}(<1)$, $N_{\max}$.

Output: RB $V = [v_1, \ldots, v_N]$.

1: Initialization: $N = 0$, $V = [\,]$, $\mu_\star = \mu_0$, $\eta(\mu_\star) = 1$.

2: **while** $\psi_N(\mu_\star) > \varepsilon_{\mathrm{ROM}}$ & $N < N_{\max}$ **do**

3: Simulate the FOM at $\mu_\star$, and adaptively select snapshots using Algorithm 4 to get

$$S^{\mathrm{A}} := \{u^{n_1}(\mu_\star), \ldots, u^{n_\ell}(\mu_\star)\}.$$

4: Perform POD process:

Compute $\bar{U}_{\mathrm{A}} := [u^{n_1}, \ldots, \bar{u}^{n_\ell}]$, $\bar{u}^j := u^j(\mu_\star) - \Pi_{\mathcal{W}_N}[u^j(\mu_\star)]$, $j = 1, \ldots, \ell$, where $\Pi_{\mathcal{W}_N}[\cdot]$ is the projection operator onto the current space $\mathcal{W}_N := \mathrm{span}\{v_1, \ldots, v_N\}$.

Compute the first POD mode v_{N+1}, the left singular vector of the matrix $\bar{U}_{\mathrm{A}}$.

5: Enrich the RB $V := [V, v_{N+1}]$.

6: $N = N + 1$.

7: Find $\mu_\star := \arg \max_{\mu \in \mathcal{P}_{\mathrm{train}}} \psi_N(\mu)$.

8: **end while**

5.3 Accelerating FOM simulation using ROMs as predictors

As shown in Algorithm 2 in Chapter 3, the RB is generated iteratively through a greedy algorithm for parametric systems. At each iteration, a FOM simulation needs to be performed at the chosen parameter to collect the snapshots for the enrichment of the current RB. It usually takes several iterations to obtain a ROM with the required accuracy. A single FOM simulation is often costly, let alone repeated simulations under parameter variations, e.g., during RB generation for the SMB chromatographic model under consideration. We next propose to accelerate the FOM simulation during the basis construction to reduce the offline cost.

For multi-stage systems, like SMB chromatography introduced in Chapter 2, it usually takes many cycles (or stages) to reach the final state, e.g., the CSS of SMB chromatography. A key observation of such kind of system is: *although the initial state does not influence the final state, it does affect the number of cycles required to achieve the final state, and in turn the computational time for achieving the final state* [116]. This implies that a "good" initial state may result in less number of cycles to reach the final state. Thus, predicting a "good" initial state is crucial to shortening the computational time of the final state.

In fact, for SMB chromatography, a cascadic multi-level method is proposed to accelerate the CSS computation in [116]. The idea of the cascadic multi-level method is to use a lower-fidelity model with a coarse spatial and/or temporal mesh to predict a CSS, and use it as the initial state on a refined mesh to obtain a CSS with desired accuracy. By doing so, the runtime for acquiring the CSS of the system can be reduced if the refinement is chosen appropriately. However, how to refine the spatial/temporal mesh, e.g., how many levels of refinement should be employed, is empirical.

We propose to use the intermediate ROMs that are generated during the RB construction process to accelerate FOM simulation for multi-stage systems such as SMB chromatography considered in this thesis. At each iteration of the RB generation process, the intermediate ROM is not yet accurate enough, but its final state (e.g., the CSS in SMB chromatography) is an approximation of that of the FOM and this state should thus be a "good" initial state for the FOM simulation in a certain sense.

During the basis construction process, a FOM simulation of the SMB model needs to

be performed at the selected parameter to acquire the snapshots for the enrichment of the current RB, as shown in Step 3 of Algorithm 2. This FOM simulation is often time-consuming, because it usually starts with a trivial initial state, e.g., zero-initial conditions. Instead of using a trivial initial condition, we use the approximated CSS solution obtained by solving the intermediate ROM constructed during the previous iteration step as the initial state to start the FOM simulation at the current iteration step in Algorithm 2. In this way, the runtime of the FOM simulation can be reduced because this "good" initial state reduces the number of switching periods required to achieve the CSS. Certainly, the reduction depends on the accuracy of the (intermediate) ROM. The more accurate the (intermediate) ROM is, the more the runtime is reduced. Since the basis construction usually takes many iterations to obtain a ROM with desired accuracy, reducing the runtime for the FOM simulations implies that the runtime for the basis construction can be largely reduced. Moreover, using a ROM as a predictor is easy to implement when the POD-Greedy algorithm is employed for the basis generation. In fact, only one step needs to be modified. That is, in Step 3 of Algorithm 2, a ROM simulation at the selected parameter is first performed, and then the FOM simulation starts from the CSS solution to the ROM rather than a trivial initial state, as shown in Algorithm 6 in the following.

Algorithm 6 RB generation for the SMB model using the POD-Greedy algorithm with ROM prediction

Input: $\mathcal{P}_{\text{train}}$, μ_0, $\varepsilon_{\text{ROM}}(< 1)$, $N_{\max}$.

Output: RB $V = [v_1, \ldots, v_N]$.

1: Initialization: $N = 0$, $\mu_\star = \mu_0$, $\psi_N(\mu_\star) = 1$, $\mathcal{V}_N = \{0\}$, $V = [\,]$.
2: **while** $\psi_N(\mu_\star) > \varepsilon_{\text{ROM}}$ & $N < N_{\max}$ **do**
3: Simulate the ROM at $\mu_\star$ to acquire its CSS solution $\hat{u}^K_{\text{CSS}}(\mu_\star)$, and simulate the FOM with $u^0(\mu_\star) = \hat{u}^K_{\text{CSS}}(\mu_\star)$ to collect snapshots $\{u^n(\mu_\star)\}_{n=0}^K$. ($u^0(\mu_\star) = 0$, a trivial initial state, when $N = 0$.)
4: Implement Steps 4–7 in Algorithm 2.
5: **end while**

Actually, the idea of using a ROM as a predictor can be found in [121], where the solution to a ROM is used as the initial state for the next iteration during an iterative process of solving a large-scale linear system ($Au = b$, $A \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $b \in \mathbb{R}^{\mathcal{N}}$) to reduce the number of the iterations. By contrast, we use the solution to the ROM as the initial state for the FOM simulation to reduce the runtime for constructing the ROM.

Chapter 6

Numerical Experiments

Contents

In this chapter, we apply the RB PMOR method to three real-life models, which are described by parametric time-dependent systems and have been introduced in Chapter 2. The derived error estimations in Chapter 4 are employed to guide the parameter sampling during the basis construction process. The accelerating techniques proposed in Chapter 5 are used to build the ROM efficiently. To show the performance of the resulting ROMs, we use the ROMs to implement the underlying optimization problems. In addition, UQ based on the ROM for nonlinear SMB chromatography is explored so that the performance of the ROM is further demonstrated. All the computations were carried out using C++ code

on a PC with an Intel Core 2 Quad CPU Q9550 2.83 GHz 4.00 GB RAM unless stated otherwise. Parts of the following results have originally presented in [187, 188, 189].

For conciseness, we adopt the following notations: PO-EB refers to the proposed primal-only output error bound in (4.6), PO-EB-f refers to the proposed primal-only error bound for the field variable in (4.12), PD-ES refers to the proposed primal-dual output error bound in (4.32).

6.1 MOR for batch chromatography

Batch chromatography is one of the major processes used for separation problems in chemical engineering, and it is described by a coupled system of convection-diffusion equations, as introduced in Chapter 2. In this section, we investigate the optimal operation of batch chromatography. The operating variable $\mu = (Q, t_{\text{in}})$ is optimally chosen in a reasonable parameter domain to maximize the production rate $Pr(\mu)$, while respecting the requirement of the recovery yield $Rec(\mu)$. Here,

$$Pr(\mu) := \frac{Qs(\mu)}{t_{\text{cyc}}}, \quad Rec(\mu) := \frac{s(\mu)}{t_{\text{in}}(c_a^{\text{f}} + c_b^{\text{f}})},$$

where

$$s(\mu) = \int_{t_3}^{t_4} c_{a,\text{O}}(t;\mu)\,\mathrm{d}t + \int_{t_1}^{t_2} c_{b,\text{O}}(t;\mu)\,\mathrm{d}t, \tag{6.1}$$

and $c_{z,\text{O}}(t;\mu) = c_z(t,1;\mu)$ is the concentration of component $z\,(z = a, b)$ at the outlet of the column. The optimization problem of batch chromatography can be formulated as follows:

$$\begin{aligned} &\min_{\mu\in\mathcal{P}}\{-Pr(\mu)\}, \\ \text{s.t.}\ \ &Rec_{\min} - Rec(\mu) \le 0, \quad \mu \in \mathcal{P}, \\ &c_z(\mu),\, q_z(\mu) \text{ are the solutions to the system (2.1)–(2.3), } z = a, b. \end{aligned} \tag{6.2}$$

It is worth noting that when solving the system (2.1)–(2.3), the time step size must be taken relatively small so that the cutting points t_i $(i = 1, \ldots, 4)$ in (6.1) are properly determined, and in turn, the related quantities $s(\mu), Pr(\mu)$, and $Rec(\mu)$ can be accurately evaluated. The small time step size results in a large number (up to $O(10^4)$) of total time

steps for every parameter $\mu \in \mathcal{P}$, which causes difficulties in the error estimation and the generation of the RB.

The model parameters and operating conditions are presented in Table 6.1. The Henry constants and thermodynamic coefficients in the isotherm equation (2.2) are given in Table 6.2. The parameter domain for the operating variable μ is $\mathcal{P} = [0.0667, 0.1667] \times [0.5, 2.0]$. The minimum recovery yield $Rec_{\min}$ is taken as 80.0%, and the purity requirements are specified as $Pu_a = 95.0\%, Pu_b = 95.0\%$, which are used to determine the cutting points t_2 and t_3 in $s(\mu)$. To capture the dynamics precisely, the dimension of spatial discretization $\mathcal{N}$ in the FOM (2.15) is taken as 1500.

Table 6.1: Model parameters and operating conditions for the batch chromatographic model.

Column dimensions (cm)	2.6×10.5
Column porosity ϵ (-)	0.4
Péclet number Pe (-)	2000
Mass-transfer coefficients κ_z, $z = a, b$ (1/s)	0.1
Feed concentrations c_z^{f}, $z = a, b$ (g/l)	2.9

Table 6.2: Coefficients of the adsorption isotherm equation for the batch chromatographic model.

H_{a1} (-)	2.69	H_{b1} (-)	3.73
H_{a2} (-)	0.1	H_{b2} (-)	0.3
K_{a1} (l/g)	0.0336	K_{b1} (l/g)	0.0446
K_{a2} (l/g)	1.0	K_{b2} (l/g)	3.0

In this section, we show the derivation of the FOM based on the finite volume discretization for the batch chromatographic model (2.1)–(2.3), and the efficient construction of the ROM. We use this example to show the performance of the derived error estimations and the ASS technique presented in the previous chapters. Finally, we show the results for the ROM-based optimization of batch chromatography.

6.1.1 ROM construction for batch chromatography

Recall that the fully discrete system for batch chromatographic model is given in (2.15). That is, the FOM reads

$$Ac_z^{n+1} = Bc_z^n + d_z^n - \frac{1-\epsilon}{\epsilon}\Delta t h_z^n, \tag{6.3}$$

$$q_z^{n+1} = q_z^n + \Delta t h_z^n, \quad z = a, b, \tag{6.4}$$

where $c_z^n := c_z^n(\mu)$, $q_z^n := q_z^n(\mu)$, $d_z^n := d_z^n(\mu), h_z^n := h_z^n(\mu) \in \mathbb{R}^{\mathcal{N}}$, $A, B \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$. Detailed description can be found in Section 2.2.1. The parameter μ characterizes the operating conditions, i.e., $\mu := (Q, t_{\text{in}})$ in this work.

We now construct a RB for each field variable. Let $N \in \mathbb{N}$ be the number of the RB vectors for c_z and q_z, and $M \in \mathbb{N}$ be the number of the CRB vectors for the operators h_a and h_b. Here for simplicity of analysis, we use the same dimension N of the RB for c_a, c_b, q_a and q_b, but one can certainly take different dimensions for the RB. This also applies to h_a and h_b. Assume that $G_z \in \mathbb{R}^{\mathcal{N}\times M}$ is the CRB for the nonlinear operator h_z, and $V_{c_z}, V_{q_z} \in \mathbb{R}^{\mathcal{N}\times N}$ $\left(V_{c_z}^T V_{c_z} = I, V_{q_z}^T V_{q_z} = I\right)$ are the RB for the field variables c_z and q_z, respectively, i.e.,

$$h_z^n \approx G_z \beta_z^n, \quad c_z^n \approx \hat{c}_z^n := V_{c_z} a_{c_z}^n, \quad q_z^n \approx \hat{q}_z^n := V_{q_z} a_{q_z}^n, \quad n = 0, \ldots, K. \tag{6.5}$$

Applying Galerkin projection and empirical operator interpolation, we formulate the ROM for the FOM (6.3)–(6.4) as follows:

$$\hat{A}_{c_z} a_{c_z}^{n+1} = \hat{B}_{c_z} a_{c_z}^n + d_0^n \hat{d}_{c_z} - \frac{1-\epsilon}{\epsilon}\Delta t \hat{H}_{c_z} \beta_z^n, \tag{6.6}$$

$$a_{q_z}^{n+1} = a_{q_z}^n + \Delta t \hat{H}_{q_z} \beta_z^n, \quad z = a, b, \tag{6.7}$$

where $a_{c_z}^n := a_{c_z}^n(\mu)$, $a_{q_z}^n := a_{q_z}^n(\mu) \in \mathbb{R}^N$ are the unknown vectors of the ROM, and $\hat{A}_{c_z} = V_{c_z}^T A V_{c_z}, \hat{B}_{c_z} = V_{c_z}^T B V_{c_z}, \hat{d}_{c_z} = V_{c_z}^T e_1$, $\hat{H}_{c_z} = V_{c_z}^T G_z$, and $\hat{H}_{q_z} = V_{q_z}^T G_z$ are the reduced matrices.

Note that $\beta_z^n := \beta_z^n(\mu) = (\beta_{z1}^n, \ldots, \beta_{zM}^n)^T \in \mathbb{R}^M$ are the vectors of coefficients for the empirical interpolation of the nonlinear operator h_z^n, and they are parameter- and time-dependent. The evaluation of β_z^n is essentially the same as the computation of the coefficients $\sigma_i(\mu)$ in (3.30) in Algorithm 3. More specifically, β_z^n are obtained by solving the

following system of equations:

$$\sum_{i=1}^{M} \beta_{z_i}^n G_{zi}(x_j) = h_{z_j}^n, \quad j = 1, \ldots, M.$$

Here, the evaluation of $h_{z_j}^n$ only needs the jth entries ($c_{a_j}^n, c_{b_j}^n$ and $q_{z_j}^n$) of the solution vectors ($c_a{}^n, c_b{}^n$ and $q_z{}^n$), i.e., $h_{z_j}^n = h_z(c_{a_j}^n, c_{b_j}^n, q_{z_j}^n)$. For the general operator empirical interpolation, the value of the operator at the interpolation point (e.g., x_j) may depend on more entries of the solution vectors (e.g., the jth entries and their neighbors) [49, 87].

As discussed in Section 3.5, the efficiency of the RB approximation is ensured by a strategy of suitable offline-online decomposition. During the offline stage, given the training sets $\mathcal{P}_{\text{train}}^{\text{crb}}$ and $\mathcal{P}_{\text{train}}$ (they can be chosen differently), Algorithm 3 is implemented to generate the CRB G_z for the nonlinear operator h_z. Then Algorithm 5 is used to generate the RB matrices V_{c_z} and V_{q_z} for c_z and q_z, respectively. As a result, all $\mathcal{N}$-dependent and μ-independent terms are precomputed and assembled to construct the reduced matrices (e.g., $\hat{A}_{c_z}, \hat{B}_{c_z}, \hat{d}_{c_z}, \hat{H}_{c_z}$, and $\hat{H}_{q_z}$). For a newly given parameter $\mu \in \mathcal{P}$, the small-sized ROM (6.6)–(6.7) is rapidly assembled and solved online so that the solution to the FOM (6.3)–(6.4) can be recovered by (6.5).

Performance of ASS To investigate the performance of the technique of ASS, we compare the runtime for RB and CRB generation with different threshold values ε_{ASS}. As shown in Algorithm 5 in Chapter 5, the ASS technique is combined with the POD-Greedy algorithm and is used for RB generation. The error indicator $\psi_N(\mu_\star)$ in Algorithm 5 involves the contribution from EI. To efficiently generate a CRB for EI, the ASS technique is also employed. The training set for CRB generation is a sample set with 25 sample points of $\mu = (Q, t_{\text{in}})$, uniformly distributed in the parameter domain. For each sample point, Algorithm 4 is used to adaptively choose the snapshots for the generation of the CRB. The runtime of CRB generation with different choices of ε_{ASS} is shown in Table 6.3. It is seen that the larger threshold ε_{ASS} is used, the more the runtime is saved. A lot of redundant information is discarded due to the adaptive selection process. Particularly, with the tolerance $\varepsilon_{\text{ASS}} = 1.0 \times 10^{-4}$, the computational time is reduced by 90.3% compared with that of the original algorithm without ASS. It should be mentioned that the tolerance ε_{ASS} cannot be taken too large; otherwise too much information from the chosen parameter will be discarded, and the parameter might be selected again afterward due to the bad approximation at this parameter. A repeated selection requires one more full simulation at

this parameter, which probably takes more time. Nevertheless, how to choose an optimal threshold is empirical and problem-dependent.

Table 6.3: Comparison of runtime of the generation of CRBs (G_a, G_b) at the same error tolerance ($\varepsilon_{\text{CRB}} = 1.0 \times 10^{-7}$) with different thresholds for ASS. $M' = 1$ is the number of the basis vectors for error estimation.

	ε_{ASS}	$\|\xi_{M+M',a}\|$	$\|\xi_{M+M',b}\|$	M (G_a G_b)		Runtime (h)
no ASS	–	9.2×10^{-8}	8.5×10^{-8}	146	152	62.50 (-)
ASS	1.0×10^{-4}	9.6×10^{-8}	8.1×10^{-8}	147	152	6.05 (−90.3%)
ASS	1.0×10^{-3}	8.7×10^{-8}	9.9×10^{-8}	147	152	3.62 (−94.2%)
ASS	1.0×10^{-2}	9.4×10^{-8}	6.2×10^{-8}	144	150	2.70 (−95.7%)

Table 6.4 shows the comparison of the runtime for RB generation by using the POD-Greedy algorithm with and without ASS. Note that the CRB is precomputed with $\varepsilon_{\text{ASS}} = 1.0 \times 10^{-4}$ for the ASS, and the corresponding runtime for CRB generation is not included here. The training set is a sample set with 60 points uniformly distributed in the parameter domain. Here and in the following, the tolerances are chosen as $\varepsilon_{\text{CRB}} = 1.0 \times 10^{-7}$, $\varepsilon_{\text{ROM}} = 1.0 \times 10^{-4}$, $\varepsilon_{\text{ASS}} = 1.0 \times 10^{-4}$. It is seen that the runtime for generating the ROM with ASS is reduced by 54.1% compared with that without ASS at the same tolerance ε_{RB}. Moreover, the accuracy of the resulting ROM with ASS is almost the same as that without ASS.

Table 6.4: Comparison of the runtime for RB generation using the POD-Greedy algorithm with and without ASS.

Algorithms	Runtime (h) [1]
POD-Greedy	14.8
ASS-POD-Greedy	6.8 (−54.1%)

[1] Due to memory limitations of the PC, this computation was done on a workstation with 4 Intel Xeon E7-8837 CPUs (8 cores per CPU) 2.67 GHz, 1 TB RAM.

It is worth noting that the ASS technique is devised for effectively collecting snapshots, and it is independent of the error indicator (true error or error bound) employed for the basis construction. Thus, the ASS is also applicable to other snapshot-based MOR methods, e.g., the POD method and the POD-DEIM method [43].

6.1.2 Output error estimation for batch chromatography

The error estimations proposed in Chapter 4 are derived for a scalar evolution equation, a single PDE. For a system of several coupled PDEs, one can analogously derive an error estimation for the underlying system by taking all the field variables as one vector. However, the behavior of the solution to each PDE might be quite different in reality. Therefore, it is desired to generate different reduced bases for each field variable, rather than using a unified basis for all the field variables.

Here, we propose to apply the error estimation to each field variable of the underlying system (2.15). Taking the primal-only error bound in (4.6) applied to for the field variable c_z as an example and recalling the detailed simulation for c_z (see (2.15)),

$$Ac_z^{n+1} = Bc_z^n + d_z^n - \frac{1-\epsilon}{\epsilon}\Delta t h_z^n, \tag{6.8}$$

the residual caused by the approximate solution $\hat{c}_z^n$ in (6.5) is

$$r_{c_z}^{n+1}(\mu) := B\hat{c}_z^n + d_z^n - \frac{1-\epsilon}{\epsilon}\Delta t \mathcal{I}_M[h_z(\hat{c}_z^n)] - A\hat{c}_z^{n+1}. \tag{6.9}$$

Observe that (6.8), (6.9) correspond to (4.1), (4.2) in the general case, respectively. Compared to the general form (4.1), the additional term d_z^n in (6.8) comes from the Neumann boundary condition, which does not depend on the solution c_z^n. Additionally, the coefficient matrices A and B are independent of time. Instead of requiring a Lipschitz continuity condition for h_z as a function of c_a^n, c_b^n and q_z^n, we assume there exists a positive constant L_h such that

$$\|h_z(c_a^n, c_b^n, q_z^n) - h_z(\hat{c}_a^n, \hat{c}_b^n, \hat{q}_z^n)\| \leq L_h \|c_z^n - \hat{c}_z^n\|, \quad n = 0, \ldots, K. \tag{6.10}$$

Assuming the initial projection error is vanishing $e_{c_z}^0(\mu) = 0$, we have a similar estimation for the approximation error $e_{c_z}^n(\mu) := c_z^n - \hat{c}_z^n$ $(n = 1, \ldots, K)$ as the following:

$$\left\|e_{c_z}^n(\mu)\right\| \leq \sum_{k=0}^{n-1} \left\|A^{-1}\right\|^{n-k} \mathtt{G}^{n-1-k} \left(\tau \epsilon_{\mathrm{EI}}^k(\mu) + \left\|r_{c_z}^{k+1}(\mu)\right\|\right), \tag{6.11}$$

where $\mathtt{G} = \|B\| + \tau L_h$, $\tau = \frac{1-\epsilon}{\epsilon}\Delta t$. More tightly,

$$\begin{aligned} \left\|e_{c_z}^n(\mu)\right\| &\leq \eta_{N,M,c_z}^n(\mu) \\ &:= \sum_{k=0}^{n-1} (\mathtt{G}_{\mathrm{F,c}})^{n-1-k} \left(\tau \left\|A^{-1}\right\| \epsilon_{\mathrm{EI}}^k(\mu) + \left\|A^{-1} r_{c_z}^{k+1}(\mu)\right\|\right), \end{aligned} \tag{6.12}$$

where $\mathtt{G}_{\mathrm{F,c}} = \left\|A^{-1}B\right\| + \tau L_h \left\|A^{-1}\right\|$.

Analogously, the error bound for the output of interest $e^n_{c_z,\mathrm{O}}(\mu) := Pc^n_z - P\hat{c}^n_z$ can be obtained based on the error bound of the field variable. Similar to (4.12), we have

$$\begin{aligned}\left\|e^{n+1}_{c_z,\mathrm{O}}(\mu)\right\| &\leq \tilde{\eta}^{n+1}_{N,M,c_z}(\mu) \\ &:= \mathtt{G}_{\mathrm{O,c}}\eta^n_{N,M,c_z}(\mu) + \tau\left\|PA^{-1}\right\|\epsilon^n_{\mathrm{EI}}(\mu) + \|P\|\left\|A^{-1}r^{n+1}_{c_z}(\mu)\right\|,\end{aligned} \tag{6.13}$$

where $\mathtt{G}_{\mathrm{O,c}} = \left\|PA^{-1}B\right\| + \tau L_h \left\|PA^{-1}\right\|$. Note that $P = (0,\ldots,0,1) \in \mathbb{R}^{\mathcal{N}}$ in this model, which means that the norm of the output $e^{n+1}_{c_z,\mathrm{O}}(\mu)$ is the absolute value of the last entry of the field variable error $e^{n+1}_{c_z}(\mu)$.

Remark 6.1.1. *The error estimate for q_a and q_b in* (2.15) *can also be obtained similarly by following the derivation of the error bound for c_z presented earlier. As the output of interest for the system in* (2.15) *only depends on c_a and c_b, the error estimations for q_a and q_b are not needed for the output error bound, and therefore are not presented here.*

Remark 6.1.2. *As mentioned earlier, it is possible to derive an error bound for the field variables $\mathbf{U} = (c_a, c_b, q_a, q_b)^T$ by considering $h_z(c_a, c_b, q_z)$ as a function of the vector $\mathbf{U}$. However, if the output error bound is derived by considering all the field variables together, the error bound for the vector $\mathbf{U}$ (denoted as $\eta^n_{N,M,\mathbf{U}}(\mu)$) will be involved, just like the error bound $\eta^n_{N,M,c_z}(\mu)$ for the field variable c_z being involved in the output error bound in* (6.13). *Obviously, the error bound $\eta^n_{N,M,\mathbf{U}}(\mu)$ is much rougher than the bound $\eta^n_{N,M,c_z}(\mu)$.*

Note that the above application of the output error bound in (4.12) by handling each field variable separately is also applied to the primal-dual output error estimation presented in Section 4.3 for batch chromatography. Moreover, for the SMB model to be presented in the following sections, the error estimation for the underlying system can be analogously derived based on that for a scalar equation derived in Chapter 4.

Performance of the primal-only error bound and the primal-dual output error estimation In the chromatographic model, given a parameter μ, the values of $Pr(\mu)$ and $Rec(\mu)$ in (6.2) are determined by the concentrations at the outlet of the column $c^n_{z,\mathrm{O}}(\mu) = Pc^n_z(\mu), n = 0,\ldots,K$, $z = a, b$, which constitute the output of the FOM in (2.15). Consequently, the output error bound will be taken as the error indicator $\psi_N(\mu)$ in the greedy algorithm (e.g., Algorithm 5) for the generation of the RB, which yields a goal-oriented ROM.

Note that the error bound $\tilde{\eta}^{n+1}_{N,M,c_z}(\mu)$ in (6.13) is the bound for the output error of the component c_z at the time instance t^{n+1} for a given parameter $\mu \in \mathcal{P}$. We use the following error indicator in Algorithm 5, $\psi_N(\mu) := \max_{\mu\in\mathcal{P}_{\text{train}}} \max_{z\in\{a,b\}} \bar{\tilde{\eta}}_{N,M,c_z}(\mu)$, where $\bar{\tilde{\eta}}_{N,M,c_z}(\mu) := \frac{1}{K}\sum_{n=1}^{K} \tilde{\eta}^n_{N,M,c_z}(\mu)$ is the average of the error bound for the output of c_z in the whole evolution process. In accordance, we define the reference true output error as $e^{\max}_N := \max_{\mu\in\mathcal{P}_{\text{train}}} \bar{e}_N(\mu)$, where $\bar{e}_N(\mu) := \max_{z\in\{a,b\}} \bar{e}_{N,c_z}(\mu)$, $\bar{e}_{N,c_z}(\mu) := \frac{1}{K}\sum_{n=1}^{K} \|c^n_{z,\mathrm{O}}(\mu) - \hat{c}^n_{z,\mathrm{O}}(\mu)\|$, and $\hat{c}^n_{z,\mathrm{O}}(\mu)$ is the approximate output response computed by the ROM in (6.6)–(6.7).

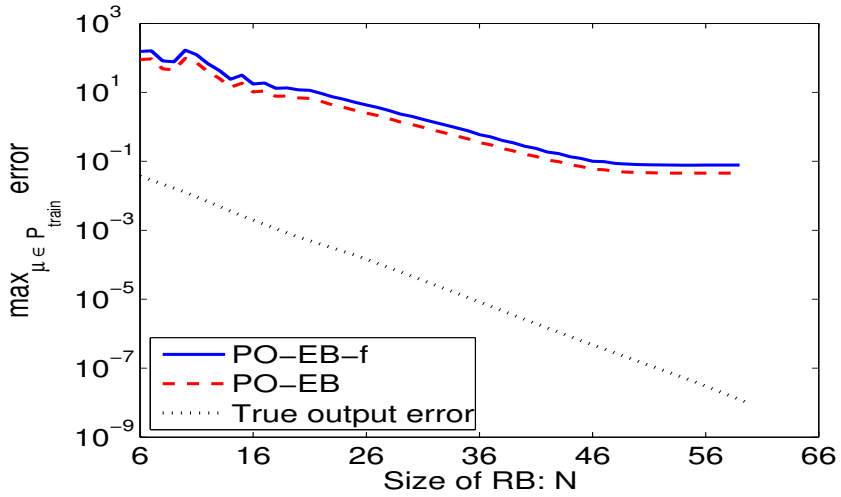


Figure 6.1: Behavior of the primal-only error bounds (PO-EB-f and PO-EB) and the true output error during the RB construction process for batch chromatography. The field variable error bound is defined as $\eta_{N,c_z} := \max_{\mu\in\mathcal{P}_{\text{train}}} \max_{z\in\{a,b\}} \{\bar{\eta}_{N,M,c_z}(\mu)\}$, where $\bar{\eta}_{N,M,c_z}(\mu) := \frac{1}{K}\sum_{n=1}^{K} \eta^n_{N,M,c_z}(\mu)$.

Figure 6.1 shows the behavior of the primal-only error bounds: the proposed primal-only output error bound (PO-EB) and the primal-only filed variable error bound (PO-EB-f), and the corresponding true output error during the RB extension using Algorithm 5. It is seen that the output error bound stagnate after certain steps, although the true error is very small already. The inefficiency of the error bound is mainly due to the accumulation of the errors over all the previous time steps. The error bound becomes less efficient when more and more previous errors are accumulated into the current error estimation over time. In such a case, a sharper error bound should be employed, as explored in

Section 4.3 and will be demonstrated in the following. Nevertheless, in the absence of better error bound, an early-stop criterion can be employed as a remedy to the stagnation resulting from overestimation of the error bound. That is, when the stagnation begins, we further check the true error at the chosen parameter. If the true error at this parameter is small enough (e.g., less than the user-specified tolerance), it is assumed that the ROM is sufficiently accurate and the iteration can be stopped. For more details, please refer to [187].

To circumvent the stagnation of PO-EB, we use the proposed primal-dual output error estimation (PD-ES) in Section 4.3. Since we have two outputs $(c_{a,\mathrm{O}}(t;\mu), c_{b,\mathrm{O}}(t;\mu))$ in this model, we define a dual system for each output to compute the individual output error estimation and take the maximum as the final error estimation for the individual, as discussed in Remark 4.3.8. Figure 6.2 shows the decay of the error estimation as the RB is enriched. It is seen that PD-ES works much better than PD-EB. PD-ES goes below the prespecified tolerance as the number of the RB increases to 45. By contrast, PO-EB almost stagnates and is still above the tolerance.

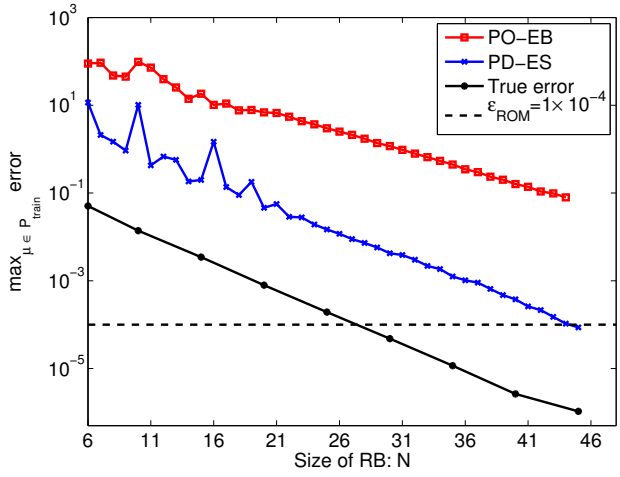


Figure 6.2: Decay of the primal-only error bound (PO-EB), the primal-dual error estimation (PD-ES), and the corresponding true error during the RB construction process for the batch chromatographic model.

To show the efficiency of PD-ES, we compare the runtime for the generation of the RB. From Table 6.5, we see that using PD-ES takes slightly more time than using PO-EB. This

is because the residual of one additional dual system needs to be computed for PD-ES. However, since PD-ES is much more accurate than PO-EB, it deserves to spend a bit more computational time for getting a more reliable ROM.

Table 6.5: Comparison of runtime for RB generation using two error estimations.

Error indicator	Runtime (h)*
PO-EB	6.8
PD-ES	7.6

* Due to memory limitations of the PC, these computations were done on a workstation with 4 Intel Xeon E7-8837 CPUs (8 cores per CPU) 2.67 GHz, 1 TB RAM.

Figure 6.3 shows the behavior of the average ratio $\tilde{\rho}_N^\star$ during the RB extension process. We have the same conclusion as that for the academic examples presented in Section 4.4, i.e., the difference between $\|\tilde{r}_{\mathrm{pr}}^{n+1}\|$ and $\|r_{\mathrm{pr}}^{n+1}\|$ becomes small as the accuracy of the ROM is increased. The ratio stays in the scale of $\mathcal{O}(1)$ when the number of basis vectors is larger than 20. This further shows that the assumption made in Corollaries 4.3.2 and 4.3.3 in Section 4.3 are reasonable.

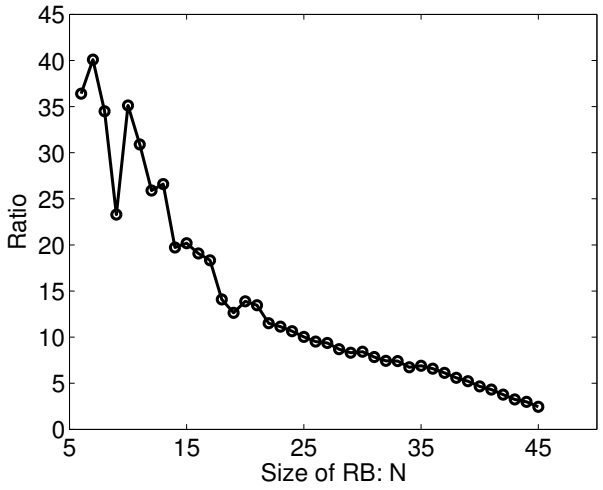


Figure 6.3: Behavior of the average ratio $\tilde{\rho}_N^\star$ during the RB construction process for the batch chromatographic model.

Figure 6.4 shows the location of the parameters selected during the RB extension process with the greedy algorithm. As mentioned in Section 3.3.2, for time-dependent problems,

one parameter value can be repeatedly selected to enrich the current RB. The size of the circle indicates how frequently the same parameter is selected for the RB extension.

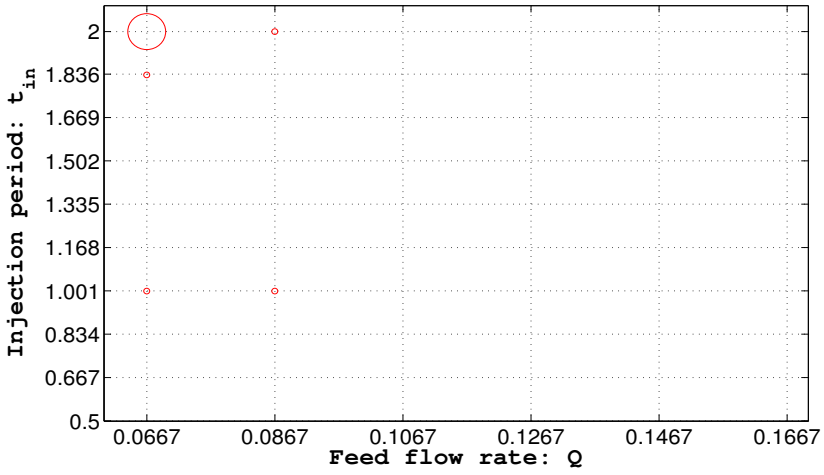


Figure 6.4: Location of the parameters selected during the RB extension process for batch chromatography. The size of a circle indicates how frequently the parameter is selected during the process. The bigger the circle is, the more often the parameter is selected.

ROM validation Before addressing the ROM based optimization, we assess the validation of the ROM. To this end, we perform full and reduced simulations over a test set with 500 random samples of the parameter in the feasible domain. Table 6.6 shows the average runtime and the maximal error over the validation set. It is seen that the average runtime is sped up by a factor of 57 using the ROM, and the maximal true output error is 8.16×10^{-7}, which is below the prespecified tolerance: $\varepsilon_{\text{ROM}} = 1 \times 10^{-4}$. In addition, the concentrations at the outlet of the column computed by using the FOM and the ROM at a given parameter $\mu = (Q, t_{\text{in}}) = (0.1018, 1.3487)$ are plotted in Figure 6.5, which shows that the ROM in (6.6)–(6.7) reproduces the dynamics of the original FOM in (6.3)–(6.4).

Table 6.6: Comparison of runtime for the full and reduced simulations of the batch chromatographic over a validation set with 500 random sample points. ($\varepsilon_{\text{ROM}} = 1 \times 10^{-4}$)

Model	Maximal error	Average runtime (s)/SpF
FOM ($\mathcal{N} = 1500$)	–	339.02 (-)
ROM ($N = 45$)	8.16×10^{-7}	5.95 / 57

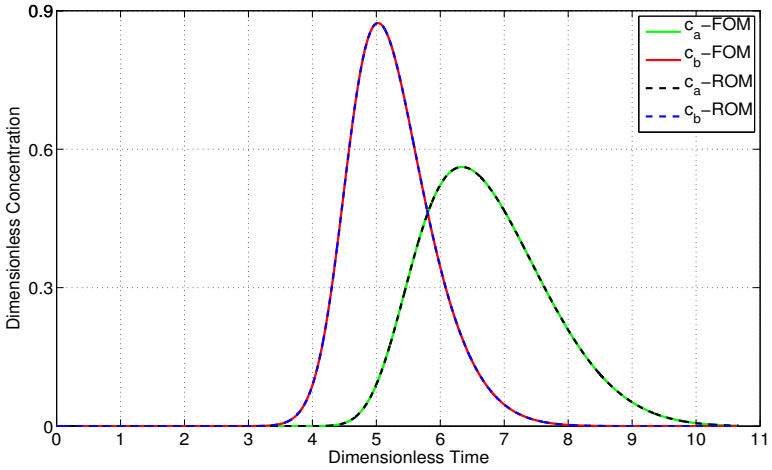


Figure 6.5: Concentrations at the outlet of the column using the FOM ($\mathcal{N} = 1500$) and the ROM ($N = 45$) at the parameter $\mu = (Q, t_{\text{in}}) = (0.1018, 1.3487)$.

6.1.3 ROM-based optimization of batch chromatography

We use the global optimizer NLOPT_GN_DIRECT_L, an efficient gradient-free algorithm in the open library NLopt [101], to solve the optimization problems. Let μ^k be the vector of parameters determined by the optimization procedure at the kth iteration, $k = 1, 2, \ldots$. When $\|\mu^{k+1} - \mu^k\| < \varepsilon_{\text{opt}}$, the optimization process is stopped and the optimal solution is obtained. The tolerance is specified as $\varepsilon_{\text{opt}} = 1.0 \times 10^{-4}$. It takes 202 iterations to reach the feasible point, and in each iteration, there are two function evaluations: one is for the objective and the other is for the constraints. The optimization results are summarized in Table 6.7. The ROM-based optimization converges to the optimal solution computed by the FOM-based optimization. Furthermore, the runtime for solving the FOM-based optimization is significantly reduced. The speedup factor (SpF) is 58.

Table 6.7: Comparison of the results for the optimization of batch chromatography based on the FOM ($\mathcal{N} = 1500$) and the ROM ($N = 45$).

Model	Objective (Pr)	Optimal solution (μ)	#Iterations	Runtime (h)/SpF
FOM	0.020264	$(0.07964, 1.05445)$	202	33.88 / -
ROM	0.020266	$(0.07964, 1.05445)$	202	0.58 / 58

6.2 MOR for linear SMB chromatography

SMB chromatography is a continuous multi-column process and has been widely used as an efficient separation technique in chemical engineering. The general description has been given in Chapter 2, where it is pointed out that different SMB processes are described by different isotherm equations. In this section, we consider an SMB model with linear isotherm equations. In the following, we first show the model and the underlying optimization problems. Then, we construct a ROM using the POD-Greedy algorithm and the RBM. Finally, we show the performance of PD-ES (the proposed primal-dual output error estimation) and the ROM-based optimization.

6.2.1 Model description and optimization

For the linear SMB model, the adsorption equilibrium in (2.5) is defined by the linear isotherm equation

$$q_{z,k}^{\mathrm{Eq}} := H_z c_{z,k}, \quad k = 1, \ldots, N_{\mathrm{col}}, \tag{6.14}$$

with H_z being the Henry constant. It is assumed that $H_a > H_b$. The model parameters are summarized in Table 6.8.

Table 6.8: Model parameters and operating conditions for the linear SMB model.

Column dimensions (cm)	2.6×11
Column porosity ϵ (-)	0.4
Péclet number Pe (-)	500
Mass-transfer coefficients κ_z, $z = a, b$ (1/s)	0.1
Feed concentrations c_z^{f}, $z = a, b$ (g/l)	2.9
Henry constants $H_a,\ H_b$ (-)	3.86, 2.72

As a case study, we use an SMB model with four zones and eight columns, as shown in Figure 2.2. In this model, four dimensionless quantities m_i, $i = \mathrm{I}, \ldots, \mathrm{IV}$, and the feed flow rate Q_{F} are chosen as the operating parameters. The four dimensionless quantities

introduced by the triangle theory [122] are defined as

$$m_i = \frac{Q_i t_\mathrm{s} - \epsilon \mathbf{V}}{\mathbf{V}(1-\epsilon)}, \quad i = \mathrm{I}, \dots, \mathrm{IV},$$

where $\mathbf{V}$ is the volume of the column. Given a set of parameters $\mu := (m_\mathrm{I}, \dots, m_\mathrm{IV}, Q_\mathrm{F})$, the SMB process reaches a CSS with periodic switching along the circularly arranged columns. The CSS condition is defined (2.14), where the CSS tolerance is taken as $\varepsilon_\mathrm{CSS} = 1 \times 10^{-4}$ in this model. The interesting parameter domain is taken as $\mathcal{P} := [4.2, 4.7] \times [2.5, 3.0] \times [3.5, 4.0] \times [2.2, 2.7] \times [0.05, 0.1]$.

In this thesis, we seek the optimal operating conditions that maximize the feed throughput while respecting the purity requirements and the process constraints. The optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{\mu \in \mathcal{P}} f(\mu) = -Q_\mathrm{F}, \\ \text{s.t.} \quad &Pu_{a,\min} - Pu_a(\mu) \leq 0, \\ &Pu_{b,\min} - Pu_b(\mu) \leq 0, \\ &Q_\mathrm{I} - Q_\max \leq 0, \end{aligned} \tag{6.15}$$

where

$$\begin{aligned} Pu_a(\mu) &:= \frac{\int_0^1 c^\mathrm{E}_{a,\mathrm{CSS}}(t,\mu)\mathrm{d}t}{\int_0^1 c^\mathrm{E}_{a,\mathrm{CSS}}(t,\mu)\mathrm{d}t + \int_0^1 c^\mathrm{E}_{b,\mathrm{CSS}}(t,\mu)\mathrm{d}t}, \\ Pu_b(\mu) &:= \frac{\int_0^1 c^\mathrm{R}_{b,\mathrm{CSS}}(t,\mu)\mathrm{d}t}{\int_0^1 c^\mathrm{R}_{a,\mathrm{CSS}}(t,\mu)\mathrm{d}t + \int_0^1 c^\mathrm{R}_{b,\mathrm{CSS}}(t,\mu)\mathrm{d}t}, \end{aligned} \tag{6.16}$$

are the product purity at the extract and the raffinate outlets, $c^\mathrm{E}_{z,\mathrm{CSS}}(t,\mu)$ and $c^\mathrm{R}_{z,\mathrm{CSS}}(t,\mu)$ are the CSS concentrations of c_z at the extract and the raffinate outlets, respectively, $z = a, b$. The constraints $Pu_{a,\min}$, $Pu_{b,\min}$, and $Q_\max$ will be specified in the numerical experiments in the following. The output is defined as

$$y(t,\mu) := (c^\mathrm{E}_{a,\mathrm{CSS}}(t;\mu), c^\mathrm{R}_{a,\mathrm{CSS}}(t;\mu), c^\mathrm{E}_{b,\mathrm{CSS}}(t;\mu), c^\mathrm{R}_{b,\mathrm{CSS}}(t;\mu)). \tag{6.17}$$

Solving such an optimization problem is time-consuming because it takes many iterations to converge and each iteration needs to simulate the original FOM until the CSS is reached. We now use MOR to tackle this problem.

6.2.2 ROM construction for linear SMB chromatography

Similarly to the nonlinear SMB model discretized in Section 2.2.2, we use the finite volume discretization to construct the FOM as follows:

$$\begin{cases} A_{\mu,z} c_z^{n+1} = B_{\mu,z} c_z^n + r_z^n + t_{\mathrm{s}} \kappa_z q_z^n, \\ q_z^{n+1} = (1 - t_{\mathrm{s}} \kappa_z \Delta t) q_z^n + t_{\mathrm{s}} \kappa_z H_z \Delta t c_z^n. \end{cases}$$

Note that the coefficient matrices $A_{\mu,z}, B_{\mu,z} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ are time independent compared with the general form in (3.22), $r_z^n \in \mathbb{R}^{\mathcal{N}}$ comes from the feed conditions, and it does not depend on the field variables, which is pretty similar to the nonlinear case presented in detail in Section 2.2.2. Let $V_{c_z} \in \mathbb{R}^{\mathcal{N} \times N_{c_z}}, V_{q_z} \in \mathbb{R}^{\mathcal{N} \times N_{q_z}}$ be the RB matrices for the field variables c_z, q_z, respectively, and $\hat{c}_z^n := V_{c_z} a_{c_z}^n$, $\hat{q}_z^n := V_{q_z} a_{q_z}^n$ be the reduced approximations of c_z^n and q_z^n, accordingly. Here $\mathcal{N}$ is the number of degrees of freedom of the FOM for each field variable, and N_{c_z}, N_{q_z} are the column numbers of the projection matrices for c_z, q_z, respectively, $z = a, b$. By using Galerkin projection, the ROM is formulated as

$$\begin{cases} \hat{A}_{\mu,z} a_{c_z}^{n+1} = \hat{B}_{\mu,z} a_{c_z}^n + \hat{r}_z + t_{\mathrm{s}} \kappa_z \hat{D}_z a_{q_z}^n, \\ a_{q_z}^{n+1} = (1 - t_{\mathrm{s}} \kappa_z \Delta t) a_{q_z}^n + t_{\mathrm{s}} \kappa_z H_z \Delta t \hat{D}_z^T a_{c_z}^n, \end{cases}$$

where $\hat{A}_{\mu,z} = V_{c_z}^T A_{\mu,z} V_{c_z}$, $\hat{B}_{\mu,z} = V_{c_z}^T B_{\mu,z} V_{c_z} \in \mathbb{R}^{N_{c_z} \times N_{c_z}}$, $\hat{r}_z = V_{c_z}^T r_z^n \in \mathbb{R}^{N_{c_z}}$ and $\hat{D}_z = V_{c_z}^T V_{q_z} \in \mathbb{R}^{N_{c_z} \times N_{q_z}}$ are the reduced matrices, and $a_{c_z}^n \in \mathbb{R}^{N_{c_z}}$, $a_{q_z}^n \in \mathbb{R}^{N_{q_z}}$ are the unknowns of the ROM, $z = a, b$.

The training set $\mathcal{P}_{\text{train}}$ consists of 150 sample points randomly distributed in the parameter domain. To generate the RB matrices V_{c_z} and V_{q_z} when using the POD-Greedy algorithm, the snapshots are taken from one CSS period rather than the transient process, since only the products in the CSS period are of interest. The number of time steps in one period is still large ($\mathcal{O}(10^3)$), which is larger than the dimension of the spatial discretization. To efficiently construct the RB, the ASS technique presented in Chapter 5 is employed. There are four outputs in this model. The similar strategy as we took for the batch chromatographic model is employed, i.e., compute an error estimation for each output and take the maximum to define the error indicator $\psi_N(\mu)$ for every parameter.

Performance of PD-ES (the primal-dual output error estimation) The column numbers of the RB matrices ($V_{c_z}, V_{q_z},\ z = a, b$) are $82, 83, 83, 83$, respectively, when the

tolerance ε_{ROM} is taken as 1.0×10^{-3}. As shown in the previous example, PD-ES outperforms PO-EB (the primal-only output error bound), thus we now only use PD-ES to construct the ROM. Figure 6.6 shows the behavior of PD-ES and the corresponding true error during the extension of the RB. PD-ES goes below the prespecified tolerance when the maximal number of the RB reaches 83. However, it does not decay smoothly, unlike the previous examples. This is because the average ratio $\tilde{\rho}_N^\star$ oscillates during the RB extension process, as illustrated in Figure 6.7. In fact, the oscillation in $\tilde{\rho}_N^\star$ will result in the oscillation in the output error estimation, since we use $\tilde{\rho}_N^\star$ to estimate $\rho^{n+1}(\mu)$ in (4.32). This is probably due to the multi-switching procedure, which causes extreme difficulty for MOR, because some error might be introduced after each switch and this error is hard to measure. In addition, the Péclet number in this model is 500, which is challenging for MOR, as we have observed for the Burgers' equation in Section 4.4.2.

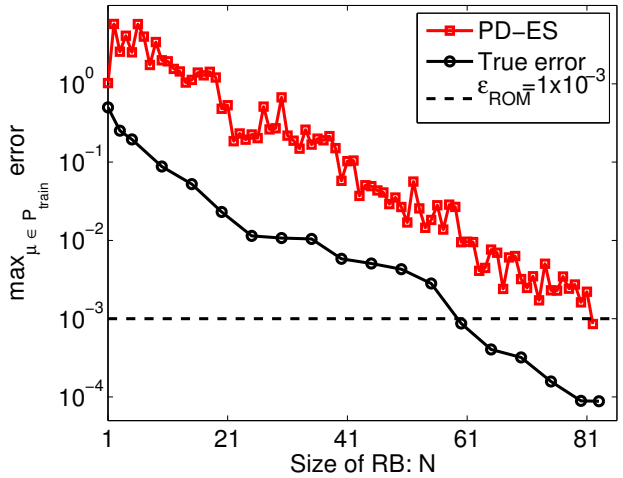


Figure 6.6: Decay of the primal-dual error estimation (PD-ES) and the corresponding true error during the RB construction process for the linear SMB model.

ROM validation Before the ROM is used to solve the underlying optimization problem, we validate its accuracy by performing the full and reduced simulation over a test set with 200 random samples of parameters in the parameter domain. The maximal error and average runtime are shown in Table 6.9. It is seen that the maximal true error is 1.1×10^{-4} and is smaller than the prespecified tolerance. The average runtime is largely reduced and the speedup factor is 7.

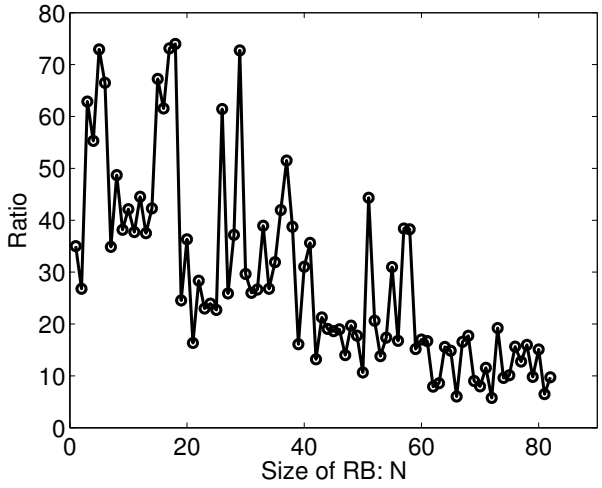


Figure 6.7: Behavior of the average ratio $\tilde{\rho}_N^\star$ during the RB construction process for the linear SMB model.

Table 6.9: Comparison of runtime for the full and reduced simulations of linear SMB model over a validation set with 200 random sample points. ($\varepsilon_{\mathrm{ROM}} = 1.0 \times 10^{-3}$)

Model	Maximal error	Average runtime (s)/SpF
FOM ($\mathcal{N} = 800$)	–	349.5 / -
ROM	1.1×10^{-4}	46.7 / 7

6.2.3 ROM-based optimization of linear SMB chromatography

To show the performance of the ROM, we implement both the FOM-based and the ROM-based optimization. Table 6.10 shows the results using the constraints $Pu_{a,\min} = 99.0\%$, $Pu_{b,\min} = 99.0\%$, $Q_{\max} = 0.50$ ml/s. We use the gradient-free optimizer, the subroutine NLOPT_LN_COBYLA [101], to solve the underlying optimization problems. It is a local optimizer. Different initial guesses may result in slightly different (locally) optimal solutions. Indeed, different initial guesses have been tested, and the differences between the optimal solution to ROM-based optimization and that to the FOM-based one are all sufficiently small.

Let μ_k be the parameter chosen by the optimizer at the kth iteration. The iteration continues until the relative variance of the decision variables μ goes below a prespecified

tolerance ε_{opt}, i.e., when $\|\mu_{k+1} - \mu_k\|/\|\mu_k\| \leq \varepsilon_{\text{opt}}$ ($\varepsilon_{\text{opt}} = 1 \times 10^{-4}$ in Table 6.10). From Table 6.10, it is seen that the ROM based optimization is very successful. The runtime for solving the optimization problem is largely reduced while the optimal solutions are almost the same as those of the FOM based optimization. The speedup factor achieved by the ROM-based optimization is 6.

Table 6.10: Comparison of the optimization results based on the FOM and the ROM.

		Initial guess	FOM	ROM
Objective	Q_{F} [ml/s]	0.07	0.0745	0.0745
Optimal solution	m_1	4.50	4.3269	4.3271
	m_2	2.90	2.8599	2.8603
	m_3	3.50	3.6036	3.6039
	m_4	2.30	2.3468	2.3685
	Q_{F} [ml/s]	0.07	0.0745	0.0745
Constraints	Pu_a	98.9%	99.0%	99.0%
	Pu_b	99.5%	99.0%	99.0%
	Q_1 [ml/s]	0.4161	0.4997	0.4998
#Iterations			71	79
Runtime (h) / SpF			5.13 / -	0.82 / 6

6.3 MOR for nonlinear SMB chromatography

Recall that the general description and numerical discretization of the nonlinear SMB model have been given in Chapter 2. In this section, we consider a binary separation process of $1,1'$-bi-2-naphthol enantiomers on cellulose triacetate, where a mixture of 72/28 (v/v) heptane/isopropanol is used as eluent [12, 112, 124], which describes a nonlinear SMB process introduced in Chapter 2. In this process, the feed concentration of each component is identical and fixed at 2.9 g/l. The maximal allowable internal flow rate $Q_{\max}$ is 1.0 ml/s. The model parameters are given in Table 6.11, and the Henry constants for the isotherm function in (2.2) are given in Table 6.12.

Next, we show the details of the ROM construction, the error behavior during the basis extension process, and the performance of the ROM in solving the optimization problem.

Table 6.11: Model parameters and operating conditions for each chromatographic column in the nonlinear SMB unit.

Number of columns N_{col}	8
Column configuration	2-2-2-2
Column dimensions (cm)	2.6×10.5
Column porosity ϵ (-)	0.4
Péclet number Pe (-)	1000
Mass-transfer coefficients κ_z, $z = a, b$ [s^{-1}]	0.1, 0.1
Feed concentrations c_z^{f}, $z = a, b$ (g/l)	2.9, 2.9

Table 6.12: Coefficients of the adsorption isotherm equations for the nonlinear SMB model.

H_{a1} (-)	2.69	H_{b1} (-)	3.73
H_{a2} (-)	0.1	H_{b2} (-)	0.3
K_{a1} (l/g)	0.0336	K_{b1} (l/g)	0.0446
K_{a2} (l/g)	1.0	K_{b2} (l/g)	3.0

Finally, we analyze the robustness of the optimal solution under flow rate uncertainty based on the resulting ROM. According to the experimental experiences and the triangle theory in [122], we choose the interesting parameter domain as $\mathcal{P} := [4.30, 4.60] \times [2.40, 2.55] \times [3.05, 3.25] \times [2.10, 2.25] \times [0.1, 0.14]$, which is used as the admissible parameter domain for the ROM construction and the optimization. The tolerances for constructing the ROM are taken as $\varepsilon_{\text{CRB}} = 1.0 \times 10^{-5}$, $\varepsilon_{\text{ROM}} = 5.0 \times 10^{-3}$, and the tolerance for the CSS condition is $\varepsilon_{\text{CSS}} = 5.0 \times 10^{-4}$.

6.3.1 ROM construction for nonlinear SMB chromatography

We now implement the RBM presented in Chapter 3 for the nonlinear SMB chromatography. For reduced-order modeling of the SMB model in (2.19)–(2.20), to reduce the order (size), we construct a RB for each variable, i.e., compute the RB matrices V_{c_z} and V_{q_z} for the variables c_z and q_z ($z = a, b$), respectively; to reduce the complexity, we con-

struct the CRB matrix G_z for the nonlinear term h_z $(z = a, b)$. Let $\hat{c}_z^n(\mu) := V_{c_z} c_{z,r}^n(\mu)$ and $\hat{q}_z^n(\mu) := V_{q_z} q_{z,r}^n(\mu)$ be the approximation to $c_z^n(\mu)$ and $q_z^n(\mu)$, respectively. Applying Galerkin projection, the ROM for the SMB model can be formulated as

$$\hat{A}_{c_z}(\mu) c_{z,r}^{n+1}(\mu) = \hat{B}_{c_z}(\mu) c_{z,r}^n(\mu) + \hat{R}_{c_z} \tilde{r}_z^n(\mu) - \frac{1-\epsilon}{\epsilon} \Delta t \hat{H}_{c_z} \beta_z^n(\mu), \tag{6.18}$$

$$q_{z,r}^{n+1} = q_{z,r}^n + \Delta t \hat{H}_{q_z} \beta_z^n(\mu), \quad z = a, b, \tag{6.19}$$

where $c_{z,r}^n(\mu)$, $q_{z,r}^n(\mu)$ are the reduced state vectors, $\beta_z^n(\mu) := \beta(c_{z,r}^n(\mu), q_{z,r}^n(\mu); \mu)$ is the vector of coefficients in the interpolation of $h_z^n(\mu)$ (i.e., $\hat{h}_z^n(\mu) = G_z \beta_z^n(\mu)$), and $\hat{A}_{c_z}(\mu) = V_{c_z}^T A(\mu) V_{c_z}, \hat{B}_{c_z}(\mu) = V_{c_z}^T B(\mu) V_{c_z}, \hat{R}_{c_z} = V_{c_z}^T e_{\wp_\mathrm{f}}$, $\hat{H}_{c_z} := V_{c_z}^T G_z$, $\hat{H}_{q_z} := V_{q_z}^T G_z$ are the reduced matrices. Here, $e_{\wp_\mathrm{f}} = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^{\mathcal{N}}$ is $\wp_\mathrm{f}$th column of the identity matrix in $\mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ ($\wp_\mathrm{f}$ corresponds to the location of the feed node of the SMB unit in the spatial grid), and $\tilde{r}_z^n(\mu)$ is actually a scalar parameter-dependent coefficient.

As discussed in Section 3.5, the RBM is usually realized by a strategy of offline-online decomposition. I.e., the construction of the ROM and the use of ROM can be completely decoupled into two stages. During the offline stage, all terms ($\hat{R}_{c_z}$, $\hat{H}_{c_z}$, $\hat{H}_{q_z}$, and the parameter-independent components, e.g., $\hat{A}_j, \hat{B}_j$ in the affine expressions of $\hat{A}_{c_z}(\mu)$ and $\hat{B}_{c_z}(\mu)$ in (3.26)) related to the high dimensional computation are precomputed and stored. This process can be expensive, but needs to be performed only once. During the online stage, given any feasible parameter, the reduced matrices ($\hat{A}_{c_z}(\mu), \hat{B}_{c_z}(\mu), \hat{R}_{c_z}$, $\hat{H}_{c_z}$, and $\hat{H}_{q_z}$) can be rapidly assembled using the precomputed data in the offline stage, and a small-sized ROM is solved. This online simulation is independent of the high dimension $\mathcal{N}$, which implies that the offline cost can be paid off by many repeated ROM simulations under parameter variations. For example, in the optimization process, both objective and constraints can be cheaply computed by using the ROM without resorting to the FOM.

We take a training set $\mathcal{P}_{\text{train}}^{\text{CRB}}$ with 150 sample points randomly distributed in the parameter domain $\mathcal{P}$ and use Algorithm 3 to construct the CRB for h_a and h_b, respectively. Then, we take a training set $\mathcal{P}_{\text{train}}^{\text{RB}}$ with 100 random samples and apply Algorithm 2 to compute the RB for each variable. The error estimator $\psi(\mu)$, presented in Section 4.3 is employed to guide the parameter sampling during the RB extension process. The corresponding true output error is defined as $e(\mu) := \frac{1}{K} \sum_{n=1}^{K} e^n(\mu)$, where $e^n(\mu) := \|y^n(\mu) - \hat{y}^n(\mu)\|_\infty$, is the true output error at time step n.

Performance of accelerating FOM simulation using ROMs as predictors As

mentioned in Chapter 5.3, the RB construction can be accelerated by using intermediate ROMs as predictors. To study the performance of the acceleration technique, we construct the RB using Algorithm 2 with and without ROM prediction. The runtime for both methods are listed in Table 6.13. It is seen that the runtime is reduced by 9.4% by using the ROM prediction. Note that the speedup is not so much although the time for the FOM simulation is reduced drastically in each iteration. This is because the time for the error estimations for all parameters in the training set dominates the whole runtime. In fact, computing the error estimation for one parameter takes roughly one ROM simulation time, which is around 0.5 minutes when the RB size N is around 40. As a result, it takes around 50 minutes to compute the error estimations for all the parameter samples (the number of parameter samples is 100). Actually, using ROMs as predictors for one FOM simulation saves around 2.6 minutes (the reduction is 65% on average), which is small compared with the total estimation time. In this case study, it takes 47 iterations to construct a ROM with the desired accuracy, and it saves more than 120 minutes in total by using ROMs as predictors. Moreover, using ROMs as predictors is entirely free, though the reduction with respect to the total time is not so high. Certainly, if much higher order FOM is employed, then the reduction by using ROMs as predictors will be much more significant because the time for FOM simulation takes more weight over the total time for constructing the ROM.

Table 6.13: Comparison of runtime for the RB construction with or without ROM prediction. ($|\mathcal{P}_{\text{train}}^{\text{RB}}| = 100$)

Methods	Runtime (h)
without ROM prediction	28.8
with ROM prediction	26.1 (-9.4%)

Performance of PD-ES (the primal-dual output error estimation) and ROM validation Figure 6.8 shows the behavior of PD-ES and the corresponding true output error during the RB construction process. It is seen that PD-ES bounds the true output error. Moreover, the estimation decays as the RB is enriched, and it goes below the tolerance ε_{ROM} when the number of basis vectors is up to 47. To further assess the reliability of the resulting ROM, we perform full and reduced simulations over a validation set $\mathcal{P}_{\text{val}}$, which consists of 200 random sample points in the parameter domain. The results are summarized in Table 6.14. It is seen that the maximal true error, $\max_{\mu\in\mathcal{P}_{\text{val}}} e(\mu)$, is

5.6×10^{-4}, which is below the prespecified tolerance. This demonstrates that the resulting ROM is reliable in the whole parameter domain. Moreover, the average runtime for one FOM simulation is 287.4 s, while it is only 28.7 s for one ROM simulation. The average speedup factor (SpF) is 10.

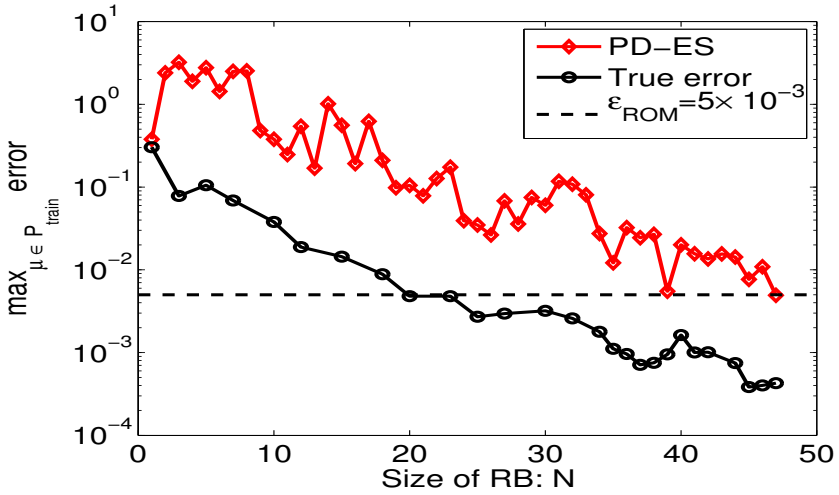


Figure 6.8: Decay of the primal-dual output error estimation (PD-ES) and the corresponding true error during the RB extension process for the nonlinear SMB model.

Table 6.14: Comparison of runtime for the full and reduced simulations of the nonlinear SMB model over a validation set with 200 random sample points. ($\varepsilon_{\text{ROM}} = 5.0 \times 10^{-3}$)

Model	Maximal error	Average runtime (s)/SpF
FOM ($\mathcal{N} = 800$)	–	287.4 / -
ROM	5.6×10^{-4}	28.7 / 10

Here, we comment on different tolerances used for the RB construction for the three models: the batch chromatographic model, the linear SMB model, and the nonlinear SMB model. The reason is that each model has its own special issues that need to be treated carefully. For example, in the batch chromatographic model, the determination of the cutting points t_j ($j = 1, \ldots, 4$) in (6.1) requires a sufficiently accurate ROM because the concentrations around these points changes drastically and tend to oscillate; in the SMB models, some error might be introduced by the periodic switching procedure, and it is hard to measure. Particularly, the nonlinear SMB model exhibits very different behaviors over a

wide range of parameter domain. All these properties make a single ROM hard to capture the dynamics with high accuracy. Therefore, without impairing the accuracy requirements from engineering applications, we take a smaller tolerance for the batch chromatographic model and a larger one for the nonlinear SMB model. Although a global ROM is employed in the current study, local basis methods might be considered in the future to improve the accuracy of the ROM.

6.3.2 ROM-based optimization of nonlinear SMB chromatography

As the ROM is precomputed in the offline stage, it is ready for online computations. When the ROM is employed for the optimization problem in (6.15), all the quantities computed through model simulations, e.g., the purity Pu_a and Pu_b, will be computed by solving the ROM in (6.18)–(6.19). Moreover, we do not modify the ROM during the online stage, since the ROM is reliable and accurate enough in the whole parameter domain.

To show the performance of the ROM, we implement both the FOM-based and the ROM-based optimization. The constraints are specified as $Q_{\max} = 1.0$ ml/s, $Pu_{a,\min} = 95.0\%, Pu_{b,\min} = 95.0\%$. The optimization problems are solved by using the optimizer NLOPT_LN_COBYLA from the NLopt library [101]. This is a local derivative-free optimizer. Different initial guesses may result in slightly different (locally) optimal solutions. Indeed, different initial guesses have been tested, and the differences between the optimal solution to the ROM-based optimization and that to the FOM-based one are all sufficiently small. The initial guess used in Table 6.15 is $\mu_0 = [m_{\mathrm{I}}, \ldots, m_{\mathrm{IV}}, Q_{\mathrm{F}}] = [4.35, 2.42, 3.21, 2.21, 0.11]$. Let μ_k be the parameter chosen by the optimizer at the kth iteration. The iteration continues until the relative variance of the decision variables μ goes below a prespecified tolerance $\varepsilon_{\mathrm{opt}}$, i.e., when $\|\mu_{k+1} - \mu_k\|/\|\mu_k\| \leq \varepsilon_{\mathrm{opt}}$ ($\varepsilon_{\mathrm{opt}} = 1 \times 10^{-3}$ in Table 6.15). The results are summarized in Table 6.15.

It is seen that the FOM-based optimization takes 102 iterations to converge and the ROM-based one takes 93 iterations. The difference between the optimal solution to the ROM-based optimization and that to the FOM-based one is acceptable. In the meanwhile, the runtime for solving the optimization is significantly reduced by using the ROM. Solving the FOM-based optimization takes 8.124 hours, while solving the ROM-based one takes only 0.786 hours. The speedup factor is 10, which is a big progress compared with the recent

Table 6.15: Comparison of the optimization results based on the FOM and the ROM.

		Initial guess	FOM	ROM
Objective	Q_{F} [ml/s]	0.11	0.1218	0.1218
Optimal solution	m_1	4.35	4.4467	4.4733
	m_2	2.42	2.4936	2.4915
	m_3	3.21	3.1163	3.1175
	m_4	2.21	2.1987	2.2000
	Q_{F} [ml/s]	0.11	0.122	0.122
Constraints	Pu_a	96.1%	95.0%	95.0%
	Pu_b	88.3%	95.0%	95.0%
	Q_1 [ml/s]	0.6985	1.0	1.0
# Iterations			102	93
Runtime (h) / SpF			8.124 / -	0.786 / 10.3

work in [112]. There, the optimization of a nonlinear SMB model is accelerated using the POD-based ROMs, and the speedup factor is around 2. There are two reasons for the improvement: one is that the ROM in [112] is locally reliable, while the proposed ROM is globally reliable in the parameter domain. As a result, the ROM in [112] has to be updated during the optimization process, which occupies much computational time. The other is that only the order of the FOM is reduced using the POD Galerkin projection method in [112], no reduction was done for the complexity of nonlinear parts, which restricts the reduction by MOR. By contrast, we have employed the RBM and the EIM to reduce both the order and the complexity of the FOM.

6.3.3 UQ of nonlinear SMB chromatography

Uncertainties in isotherm parameters, pump stability and calibration, extra-column volumes, and packing reproducibility, are inevitable in every SMB process [127]. There, the authors proposed an optimal design method for a linear SMB process under flow rate uncertainty. In this work, we use the Monte-Carlo method [88], the standard UQ method, to analyze the robustness of the product purity under flow rate uncertainty for a nonlinear SMB process.

Table 6.16: UQ for uncertainty of the flow rate Q_{I}.

Model	Order	CPU time (h)	$\mathcal{E}[Pu_a]$ [%]	$\mathcal{E}[Pu_b]$ [%]	$\delta[Pu_a]$ [%]	$\delta[Pu_b]$ [%]
FOM	800	2.04	95.00	94.96	0.07	0.15
ROM	47	0.20	94.99	94.96	0.07	0.15

Table 6.17: UQ for uncertainty of the flow rate Q_{II}.

Model	Order	CPU time (h)	$\mathcal{E}[Pu_a]$ [%]	$\mathcal{E}[Pu_b]$ [%]	$\delta[Pu_a]$ [%]	$\delta[Pu_b]$ [%]
FOM	800	2.00	94.80	94.96	0.65	0.54
ROM	47	0.20	94.80	94.96	0.65	0.54

To analyze the influence of the flow rate uncertainty in each zone upon the product purity, the flow rate in a certain zone is allowed to undertake a $\pm 2\%$ deviation while those in the other zones are fixed. Note that the mean value of the varying flow rate and the fixed value of the flow rate are the optimal solution obtained by the ROM-based optimization, which is $(Q_{\mathrm{I}}, \ldots, Q_{\mathrm{IV}}) = (0.99, 0.6144, 0.7362, 0.5578)$ [ml/s]. The switching period is fixed at $t_s = 172$ s. For each case, 25 groups of random samples of the parameters are taken for the UQ based on the FOM and the ROM, respectively. More specifically, for the FOM-based UQ, we solve the FOM in (2.19)–(2.20) at each group of the sample to compute the output $y(t^n, \mu)$ and in turn the product purity in (6.16). For the ROM-based UQ, we solve the ROM at the same group of parameter samples to compute the corresponding quantities approximately.

The statistical quantities, e.g., the mean value $\mathcal{E}[\cdot]$ and the standard deviation $\delta[\cdot]$ of the product purity, are presented in Tables 6.16–6.19 for the varying flow rate in the four zones, respectively. It is observed that the purity of the component a is more sensitive to Q_{II}, reflected by larger variance of $\delta[Pu_a]$ in Table 6.17. The purity of the component b is more sensitive to Q_{III}, reflected by larger variance of $\delta[Pu_b]$ in Table 6.18. By contrast, Tables 6.16 and 6.19 show that the flow rates Q_{I} and Q_{IV} have less effect on the purity of

Table 6.18: UQ for uncertainty of the flow rate Q_{III}.

Model	Order	CPU time (h)	$\mathcal{E}[Pu_a]$ [%]	$\mathcal{E}[Pu_b]$ [%]	$\delta[Pu_a]$ [%]	$\delta[Pu_b]$ [%]
FOM	800	1.94	94.78	94.92	0.49	2.20
ROM	47	0.19	94.78	94.92	0.49	2.20

Table 6.19: UQ for uncertainty of the flow rate Q_{IV}.

Model	Order	CPU time (h)	$\mathcal{E}[Pu_a]$ [%]	$\mathcal{E}[Pu_b]$ [%]	$\delta[Pu_a]$ [%]	$\delta[Pu_b]$ [%]
FOM	800	2.03	94.93	95.04	0.02	0.03
ROM	47	0.20	94.92	95.04	0.03	0.04

both products, since the variance are all much smaller. In addition, the statistic quantities (the mean and the standard derivation of the product purity) obtained from the ROM are almost the same as those obtained from the FOM. The runtime is significantly reduced using the ROM, and the SpF for all cases is around 10.

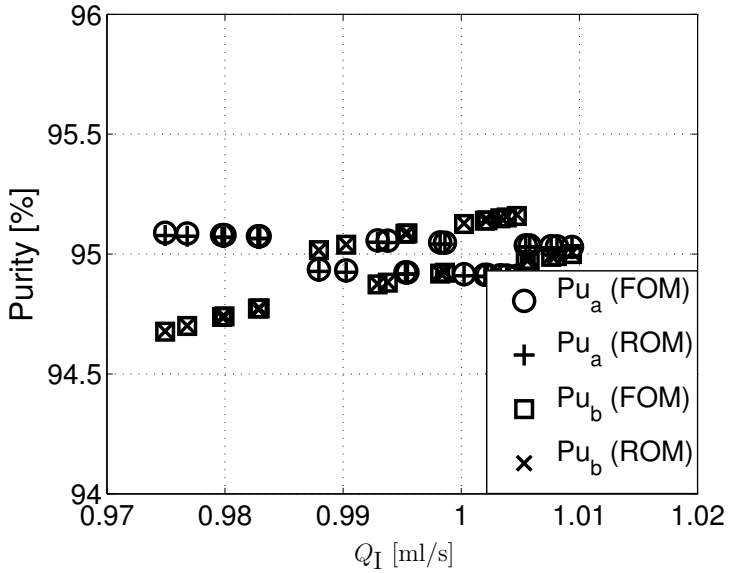


Figure 6.9: Effect of Q_{I} on the product purity Pu_a and Pu_b using the FOM and the ROM, respectively.

Figures 6.9–6.12 show the profiles of the product purity corresponding to the varying flow rates in zone I, II, III, and IV, respectively. It is shown that the results computed by using the ROM have the same behavior as those of the FOM. Figure 6.9 shows that the purity of both products does not change monotonically as the flow rate in zone I increases (or decreases), unlike the behavior of a linear SMB model presented in [127], where the changes of the purity with respect to the flow rate are all monotonic. This reflects the complex nonlinear relation between the purity and the flow rate in the nonlinear SMB model. It is also noteworthy from Figure 6.11 that as the flow rate Q_{III} increases, the purity of

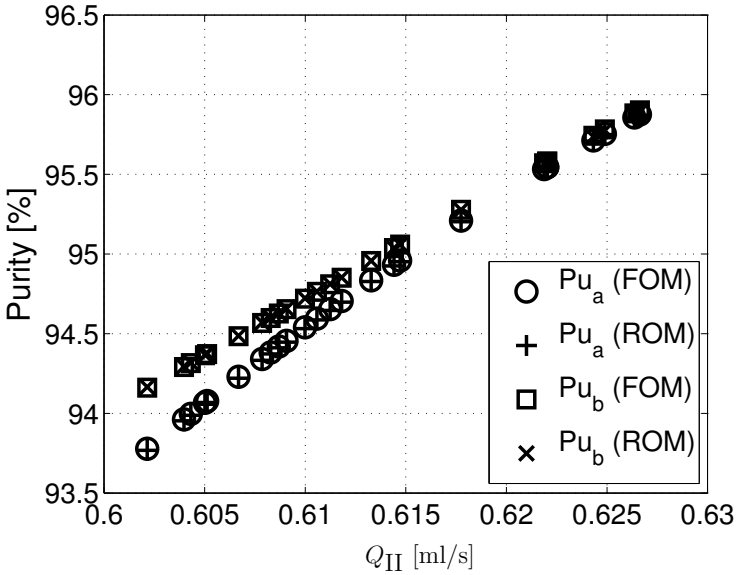


Figure 6.10: Effect of Q_{II} on the product purity Pu_a and Pu_b using the FOM and the ROM, respectively.

the component b becomes higher while the purity of the component a becomes lower. Moreover, only one value of Q_{III} (provided that the other conditions are fixed) satisfies the purity requirement of both components, which is exactly the optimal solution.

In summary, the optimal solution is still reliable if the deviations of the flow rates in zone II and III are relatively small. From Figures 6.10 and 6.11, if the purity is allowed to deviate $\pm 0.5\%$ from the mean value 95.0%, the purified products are still acceptable as long as the deviations of the flow rates in zones II and III are less than 1.0%. Nevertheless, the flow rates in zone I and IV have less influence on the product purity, as can be seen from Figures 6.9 and 6.12. Through the UQ, it is further demonstrated that the parametric ROM is qualified for the many-query task.

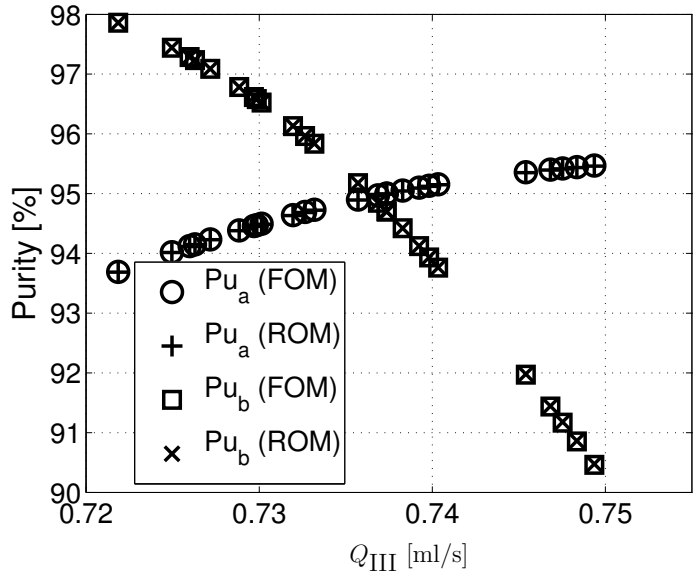


Figure 6.11: Effect of Q_{III} on the product purity Pu_a and Pu_b using the FOM and the ROM, respectively.

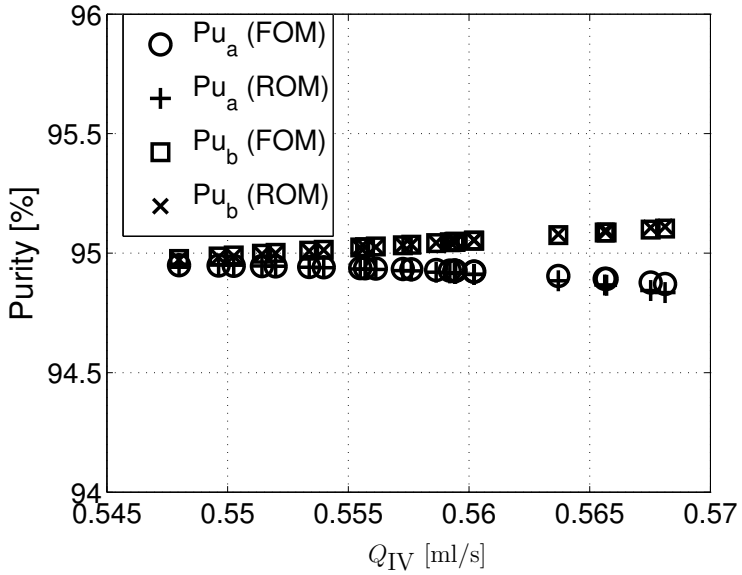


Figure 6.12: Effect of Q_{IV} on the product purity Pu_a and Pu_b using the FOM and the ROM, respectively.

Chapter 7

Conclusions and Perspectives

Contents

This chapter concludes the thesis, summarizes its contributions, and shows some perspectives on the future work.

7.1 Conclusions

In this thesis, we have explored MOR for parametric nonlinear time-dependent problems. Motivated by real-life models from chemical engineering, we start with a compact review of commonly used MOR methods, and then choose the RBM as the tool to deal with the underlying problems. In particular, we focus on error estimation and efficient RB generation for constructing ROMs. Some new error estimations and accelerating strategies are proposed for PMOR of parameter- and time-dependent problems. The proposed methods are not only tested by academic examples but also applied to real-life models. In addition, ROM-based optimization and ROM-based UQ of batch and SMB chromatography have been explored. The main contributions can be summarized as follows:

1. Two output error estimations are derived for PMOR of parameterized nonlinear evolution equations.

2. The ASS technique is proposed to reduce the cost of ROM construction by effectively collecting the snapshots, and it is suitable for snapshot-based MOR methods applied to time-dependent problems.

3. Accelerating FOM simulation by using ROMs as predictors is proposed for multi-stage systems.

4. ROM-based optimization of batch and SMB chromatography is successfully implemented, and the significance of the reduction in computational cost by using the parametric ROMs is demonstrated.

5. ROM-based UQ of nonlinear SMB chromatography is preliminarily explored, and the robustness of the separation process is efficiently analyzed by using the parametric ROM.

As mentioned in the introduction, this thesis is aimed at deriving efficient and sharp error bounds/estimation and seeking better strategies for the generation of parametric ROMs that are reliable in a wide range of the parameter values. In Chapter 4, we derived a primal-only output error bound based on the analysis of the residual of the primal system in a fully discrete framework of projection-based PMOR for evolution equations. Although the derived error bound is efficient, it may lose its sharpness for some problems especially when a large number of time steps are needed. To circumvent the problem, we proposed a sharper primal-dual output error bound by introducing and using a novel dual approach. However, an estimation needs to be performed for practical computation of the primal-dual error bound. As a result, it reduces to an output error estimation. The error bound/estimation are derived algebraically in the finite-dimensional vector space so that they are independent of the spatial discretization method employed. Moreover, they are applicable to various (P)MOR methods for estimating the error in the time domain, as reviewed in Chapter 3. Results of the academic numerical examples show that the proposed error estimations are applicable to a broad class of parameterized evolution equations.

In Chapter 5, we addressed how to efficiently construct ROMs for certain kinds of problems, and proposed two accelerating techniques for parameterized time-dependent problems. For problems that require a large number of time steps, we proposed the ASS technique to collect the snapshots effectively by discarding the redundant information according to the solution variations within the time trajectory for a given parameter. With ASS, the

offline time can be largely reduced, while the accuracy of the ROM can be guaranteed. For multi-stage systems, like the SMB process considered in this thesis, the offline cost can be further reduced by accelerating FOM simulation using (intermediate) ROMs as predictors during the RB construction process, as shown in Section 6.3.1.

The performance of the parametric ROMs is demonstrated in time-critical applications to real-life models in chemical engineering. ROM-based optimization of batch and SMB chromatography and UQ of a nonlinear SMB model have been explored, as detailed in Chapter 6.

7.2 Future work

Many problems remain to be investigated in the future and are specified as follows. First, the derived primal-dual output error bound is applicable to a broad class of evolution equations, and it is fairly sharp and efficient. However, since the quantity $\rho^n(\mu)$ in the primal-dual error bound needs to be estimated, the rigorousness of the error estimate cannot be guaranteed, though loss of the "upper bound" property was not observed for the examples tested in this thesis. A convincing and more reliable estimation of $\rho^n(\mu)$ in the primal-dual error bound deserves further investigation.

Second, the proposed error estimations are derived based on a quasi-implicit scheme, where the nonlinear parts are computed by using an explicit scheme. The extension to a more general case, i.e., computing nonlinear parts by using an implicit scheme, is also possible, though it is not straightforward. In that case, the resulting system of nonlinear equations can be solved, e.g., by the Newton method. Nevertheless, the Jacobian should be efficiently computed and the error estimation should be re-studied, which deserve further investigation.

Third, when (P)MOR is applied to physical models, e.g., problems in fluid dynamics and chemical engineering, special attention should be paid to certain quantities with physical meaning, e.g., the pressure, the density, the concentration, and those that should be positive or satisfy some critical conditions. In fact, many strategies (e.g., methods based on the (local) maximum principle) have been proposed to preserve the positivity in solving the high-fidelity model. However, when MOR is applied, there is no physical meaning in

the reduced state vector, and it is difficult to guarantee the positivity of the (prolongated) reduced approximation. Actually, for time-dependent problems, it is impractical to check whether the positivity of the reduced approximation is preserved at each time instance because the computation of the reduced approximation needs to prolongate the reduced state from the (reduced) low-dimensional space back to the (original) high-dimensional space and this is time-consuming. However, for special systems, e.g., the multi-stage systems like the SMB model considered in this thesis, it might be possible to check it at the end of each stage (or period). Further investigation is desired.

Last but not least, to deal with the nonlinearity in the model, the EIM or a similar technique can be employed. That is, an additional basis, e.g., the CRB for the EIM, is constructed for the interpolation, and this basis is usually constructed before the generation of RB for the field variable, as discussed in Section 3.4. Moreover, to ensure the accuracy of the interpolation, the dimension of the CRB is often simply taken very high. This is, however, unnecessary because the error introduced by the projection dominates the total error when the dimension of the RB is relatively low. Thus, how to well balance the dimensions of the two bases (i.e., the RB and the CRB) is also an interesting topic for research.

Bibliography

[1] N. M. Alexandrov, J. E. Dennis Jr, R. M. Lewis, and V. Torczon, *A trust-region framework for managing the use of approximation models in optimization*, Struct. Optim., 15 (1998), pp. 16–23. (cf. p. 22)

[2] N. M. Alexandrov and R. M. Lewis, *An overview of first-order model management for engineering optimization*, Optim. Eng., 2 (2001), pp. 413–430. (cf. p. 22)

[3] B. O. Almroth, P. Stern, and F. A. Brogan, *Automatic choice of global shape functions in structural analysis*, AIAA J., 16 (1978), pp. 525–528. (cf. p. 33)

[4] D. Amsallem and C. Farhat, *Stabilization of projection-based reduced-order models*, Internat. J. Numer. Methods Engrg., 91 (2012), pp. 358–377. (cf. p. 39)

[5] D. Amsallem, M. J. Zahr, Y. Choi, and C. Farhat, *Design optimization using hyper-reduced-order models*, Struct. Multidiscip. Optim., 51 (2015), pp. 919–940. (cf. p. 23)

[6] D. Amsallem, M. J. Zahr, and C. Farhat, *Nonlinear model order reduction based on local reduced-order bases*, Internat. J. Numer. Methods Engrg., 92 (2012), pp. 891–916. (cf. p. 43)

[7] D. Amsallem, M. J. Zahr, and K. Washabaugh, *Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction*, Adv. Comput. Math., (2015), pp. 1–44. (cf. p. 43)

[8] H. Antil, M. Heinkenschloss, R. H. Hoppe, and D. C. Sorensen, *Domain decomposition and model reduction for the numerical solution of PDE constrained optimization problems with localized optimization variables*, Comput. Vis. Sci., 13 (2010), pp. 249–264. (cf. p. 23)

[9] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM Publications, Philadelphia, PA, 2005. (cf. pp. 4, 28, 30, 35)

[10] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, *A survey of model reduction methods for large-scale systems*, Contemp. Math., 280 (2001), pp. 193–219. (cf. p. 35)

[11] J. M. M. Araújo, R. C. R. Rodrigues, and J. P. B. Mota, *Optimal design and operation of a certain class of asynchronous simulated moving bed processes*, J. Chromatogr. A, 1132 (2006), pp. 76–89. (cf. pp. 4, 20)

[12] ——, *Use of single-column models for efficient computation of the periodic state of a simulated moving-bed process*, Ind. Eng. Chem. Res., 45 (2006), pp. 5314–5325. (cf. pp. 4, 13, 20, 102)

[13] P. Astrid, S. Weiland, K. Willcox, and T. Backx, *Missing point estimation in models described by proper orthogonal decomposition*, IEEE Trans. Autom. Control, 53 (2008), pp. 2237–2251. (cf. p. 49)

[14] Z. Bai, *Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems*, Appl. Numer. Math., 43 (2002), pp. 9–44. (cf. pp. 3, 30, 59)

[15] Z. Bai and D. Skoogh, *A projection method for model reduction of bilinear dynamical systems*, Linear Algebra Appl., 415 (2006), pp. 406–425. (cf. p. 34)

[16] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza, *Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations*, Internat. J. Numer. Methods Engrg., 102 (2015), pp. 1136–1161. (cf. p. 39)

[17] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672. (cf. pp. 5, 32, 38, 44, 45, 46, 48, 65)

[18] J.-F. Barthelemy and R. T. Haftka, *Approximation concepts for optimum structural design — a review*, Struct. Multidiscip. Optim., 5 (1993), pp. 129–144. (cf. p. 22)

[19] U. Baur, P. Benner, and L. Feng, *Model order reduction for linear and nonlinear systems: a system-theoretic perspective*, Arch. Comput. Methods Eng., 21 (2014), pp. 331–358. (cf. pp. 3, 4, 28, 29, 30, 35)

[20] U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Maier, and M. Ohlberger, *Comparison of methods for parametric model order reduction of instationary problems*, Preprint MPIMD/15-01, Max Planck Institute Magdeburg, Feb. 2015. Available from `http://www.mpi-magdeburg.mpg.de/preprints/`. (cf. p. 28)

[21] M. Bebendorf, Y. Maday, and B. Stamm, *Comparison of some reduced representation approximations*, in Reduced Order Methods for Modeling and Computational Reduction, MS&A Series, A. Quarteroni and G. Rozza, eds., vol. 9, Springer, Cham, Switzerland, 2014, pp. 159–185. (cf. p. 49)

[22] T. Bechtold, G. Schrag, and L. Feng, eds., *System-level Modeling of MEMS*, Advanced Micro & Nanosystems, Wiley-VCH, 2013. (cf. pp. 4, 35)

[23] P. Benner and T. Breiten, *Krylov-subspace based model reduction of nonlinear circuit models using bilinear and quadratic-linear approximations*, in Progress in Industrial Mathematics at ECMI 2010, Mathematics in Industry, M. Günther, A. Bartel, M. Brunk, S. Schöps, and M. Striebel, eds., vol. 17, Springer-Verlag, Berlin, 2012, pp. 153–159. (cf. p. 34)

[24] P. Benner and L. Feng, *A robust algorithm for parametric model order reduction based on implicit moment matching*, in Reduced Order Methods for Modeling and Computational Reduction, MS&A Series, A. Quarteroni and G. Rozza, eds., vol. 9, Springer, Cham, Switzerland, 2014, pp. 159–185. (cf. pp. 3, 30, 59)

[25] P. Benner, L. Feng, S. Li, and Y. Zhang, *Reduced-order modeling and ROM-based optimization of batch chromatography*, in Numerical Mathematics and Advanced Applications-ENUMATH 2013, A. Abdulle, S. Deparis, D. Kressner, F. Nobile, and M. Picasso, eds., vol. 103 of Lect. Notes Comput. Sci. Eng., Springer, Cham, Switzerland, 2015, pp. 427–435. (cf. pp. , 46, 79, 80)

[26] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531. (cf. pp. 4, 23, 28, 35, 39, 43)

[27] P. Benner, V. Mehrmann, and D. Sorensen, *Dimension Reduction of Large-Scale Systems*, vol. 45 of Lect. Notes Comput. Sci. Eng., Springer-Verlag, Berlin/Heidelberg, Germany, 2005. (cf. pp. 4, 35)

[28] P. Benner, E. Sachs, and S. Volkwein, *Model order reduction for PDE constrained optimization*, in Trends in PDE Constrained Optimization, G. Leugering, P. Benner, S. Engell, A. Griewank, H. Harbrecht, M. Hinze, R. Rannacher, and S. Ulbrich, eds., vol. 165 of International Series of Numerical Mathematics, Birkhäuser, Basel, Switzerland, 2014, pp. 303–326. (cf. p. 22)

[29] L. T. Biegler, O. Ghattas, M. Heinkenschloss D. Keyes, and B. van Bloemen Waanders, eds., *Real-Time PDE-Constrained Optimization*, SIAM Publications, Philadelphia, PA, 2007. (cf. p. 22)

[30] L. T. Biegler, O. Ghattas, M. Heinkenschloss and B. van Bloemen Waanders, eds., *Large-scale PDE-constrained Optimization*, vol. 30 of Lect. Notes Comput. Sci. Eng., Springer-Verlag, Berlin, 2003. (cf. p. 22)

[31] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk, *Convergence rates for greedy algorithms in reduced basis methods*, SIAM J. Math. Anal., 43 (2011), pp. 1457–1472. (cf. p. 43)

[32] D. B. Broughton and C. G. Gerhold, *Continuous sorption process employing fixed bed of sorbent and moving inlets and outlets*, US Patent 2,985,589, May 23 1961. (cf. p. 13)

[33] A. Buffa, Y. Maday, A. T. Patera, C. Prudhomme, and G. Turinici, *A priori convergence of the greedy algorithm for the parametrized reduced basis method*, ESAIM: Math. Model. Numer. Anal., 46 (2012), pp. 595–603. (cf. p. 43)

[34] T. Bui-Thanh, *Model-constrained optimization methods for reduction of parameterized large-scale systems*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2007. (cf. pp. 6, 41, 78)

[35] T. Bui-Thanh, M. Damodaran, and K. Willcox, *Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics*, AIAA Paper, 4213 (2003). (cf. pp. 3, 35, 59)

[36] T. Bui-Thanh, M. Damodaran, and K. Willcox, *Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition*, AIAA J., 42 (2004), pp. 1505–1516. (cf. p. 49)

[37] T. Bui-Thanh, K. Willcox, and O. Ghattas, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM J. Sci. Comput., 30 (2008), pp. 3270–3288. (cf. p. 39)

[38] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders, *Goal-oriented, model-constrained optimization for reduction of large-scale systems*, J. Comput. Phys., 224 (2007), pp. 880–896. (cf. p. 32)

[39] C. Canuto, T. Tonn, and K. Urban, *A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear PDEs*, SIAM J. Numer. Anal., 47 (2009), pp. 2001–2022. (cf. pp. 6, 52)

[40] K. Carlberg, *Model Reduction of Nonlinear Mechanical Systems via Optimal Projection and Tensor Approximation*, PhD thesis, Stanford University, 2011. (cf. pp. 22, 32)

[41] K. Carlberg, C. Bou-Mosleh, and C. Farhat, *Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations*, Internat. J. Numer. Methods Engrg., 86 (2011), pp. 155–181. (cf. pp. 45, 49)

[42] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, *The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows*, J. Comput. Phys., 242 (2013), pp. 623–647. (cf. p. 49)

[43] S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764. (cf. pp. 32, 38, 44, 45, 48, 59, 89)

[44] S. Chaturantabut and D. C. Sorensen, *Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 337–353. (cf. p. 32)

[45] ——, *A state space error estimate for POD-DEIM nonlinear model reduction*, SIAM J. Numer. Anal., 50 (2012), pp. 46–63. (cf. pp. 5, 48, 52)

[46] E. A. Christensen, M. Brøns, and J. N. Sørensen, *Evaluation of proper orthogonal decomposition–based decomposition techniques applied to parameter-dependent nonturbulent flows*, SIAM J. Sci. Comput., 21 (1999), pp. 1419–1434. (cf. pp. 32, 35)

[47] W. Dahmen, C. Huang, C. Schwab, and G. Welper, *Adaptive Petrov–Galerkin methods for first order transport equations*, SIAM J. Numer. Anal., 50 (2012), pp. 2420–2445. (cf. p. 33)

[48] M. A. Dihlmann and B. Haasdonk, *Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems*, Comput. Optim. Appl., 60 (2015), pp. 753–787. (cf. p. 23)

[49] M. Drohmann, B. Haasdonk, and M. Ohlberger, *Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. 937–969. (cf. pp. 5, 6, 34, 38, 42, 48, 49, 52, 53, 56, 57, 58, 88)

[50] G. Dünnebier, J. Fricke, and K.-U. Klatt, *Optimal design and operation of simulated moving bed chromatographic reactors*, Ind. Eng. Chem. Res., 39 (2000), pp. 2290–2304. (cf. pp. 4, 20)

[51] G. Dünnebier and K.-U. Klatt, *Optimal operation of simulated moving bed chromatographic processes*, Comput. Chem. Eng., 23 (1999), pp. S195–S198. (cf. pp. 4, 20)

[52] J. L. Eftang, M. A. Grepl, and A. T. Patera, *A posteriori error bounds for the empirical interpolation method*, C. R. Math. Acad. Sci. Paris, 348 (2010), pp. 575–579. (cf. p. 47)

[53] J. L. Eftang, D. J. Knezevic, and A. T. Patera, *An hp certified reduced basis method for parametrized parabolic partial differential equations*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 395–422. (cf. pp. 6, 41, 78)

[54] R. Eid, *Time Domain Model Reduction by Moment Matching*, PhD thesis, Universität München, 2009. (cf. p. 34)

[55] M. Eldred and D. Dunlavy, *Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models*, in Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA Paper 2006-7117, Portsmouth, VA, vol. 199, 2006. (cf. p. 22)

[56] M. S. Eldred, A. A. Giunta, and S. S. Collis, *Second-order corrections for surrogate-based optimization with model hierarchies*, in Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA Paper 2004-4457, Albany, NY, 2004. (cf. p. 22)

[57] R. Everson and L. Sirovich, *Karhunen–Loève procedure for gappy data*, J. Opt. Soc. Am., 12 (1995), pp. 1657–1664. (cf. p. 49)

[58] M. Fahl and E. W. Sachs, *Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition*, in Large-scale PDE-constrained optimization (Santa Fe, NM, 2001), vol. 30 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2003, pp. 268–280. (cf. p. 23)

[59] C. Farhat, T. Chapman, and P. Avery, *Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models*, Internat. J. Numer. Methods Engrg., 102 (2015), pp. 1077–1110. (cf. p. 39)

[60] P. Feldmann and R. W. Freund, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 14 (1995), pp. 639–649. (cf. p. 3)

[61] L. Feng, A. C. Antoulas, and P. Benner, *Some a posteriori error bounds for reduced order modelling of (non-)parametrized linear systems*, Preprint MPIMD/15-17, Max Planck Institute Magdeburg, Oct. 2015. Available from `http://www.mpi-magdeburg.mpg.de/preprints/`. (cf. p. 59)

[62] L. Feng, P. Benner, and A. C. Antoulas, *Automatic generation of reduced order models for linear parametric systems*, Progress in Industrial Mathematics at ECMI 2014, Mathematics in Industry, (2014). (cf. pp. 30, 40, 59)

[63] L. Feng, P. Benner, and J. G. Korvink, *System-Level Modeling of MEMS by Means of Model Order Reduction (Mathematical Approximation) - Mathematical Background*, in System-level Modeling of MEMS, T. Bechtold, G. Schrag, and

L. Feng, eds., vol. 10 of Advanced Micro and Nanosystems, Wiley-VCH, Weinheim, 2013, pp. 53–93. (cf. pp. 30, 35)

[64] M. Frangos, Y. Marzouk, K. Willcox, and B. van Bloemen Waanders, *Surrogate and reduced-order modeling: A comparison of approaches for large-scale statistical inverse problems*, in Large-Scale Inverse Problems and Quantification of Uncertainty, L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, Y. Marzouk, B. van Bloemen Waanders, and K. Willcox, eds., John Wiley & Sons, Ltd., 2010, pp. 123–149. (cf. p. 23)

[65] R. W. Freund, *Model reduction methods based on Krylov subspaces*, Acta Numer., 12 (2003), pp. 267–319. (cf. pp. 3, 29, 30, 59)

[66] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas, *Non-linear model reduction for uncertainty quantification in large-scale inverse problems*, Internat. J. Numer. Methods Engrg., 81 (2010), pp. 1581–1608. (cf. p. 47)

[67] W. Gao and S. Engell, *Iterative set-point optimization of batch chromatography*, Comput. Chem. Eng., 29 (2005), pp. 1401–1409. (cf. pp. 4, 20)

[68] K. Glover, *All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error norms*, Internat. J. Control, 39 (1984), pp. 1115–1193. (cf. p. 28)

[69] M. A. Grepl, *Reduced-basis approximation a posteriori error estimation for parabolic partial differential equations*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005. (cf. pp. 33, 37, 47, 52, 58, 67, 68)

[70] M. A. Grepl, *Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations*, Math. Models Methods Appl. Sci., 22 (2012). (cf. pp. 33, 46, 47, 52)

[71] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, M2AN Math. Model. Numer. Anal., 41 (2007), pp. 575–605. (cf. pp. 46, 47, 49)

[72] M. A. Grepl and A. T. Patera, *A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations*, M2AN Math. Model. Numer. Anal., 39 (2005), pp. 157–181. (cf. pp. 6, 33, 52, 58, 67, 68)

[73] E. J. Grimme, *Krylov projection methods for model reduction*, PhD thesis, Univ. of Illinois at Urbana-Champaign, USA, 1997. (cf. pp. 3, 29, 30)

[74] D. Gromov, S. Li, and J. Raisch, *A hierarchical approach to optimal control of a hybrid chromatographic batch process*, in Advanced Control of Chemical Processes, vol. 7, 2009, pp. 339–344. (cf. pp. 4, 20)

[75] C. Gu, *QLMOR: A new projection-based approach for nonlinear model order reduction*, in IEEE Trans. Comput.-Aided Design-Digest of Technical Papers, ICCAD 2009, Nov. 2009, pp. 389–396. (cf. p. 34)

[76] ——, *QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 30 (2011), pp. 1307–1320. (cf. p. 34)

[77] S. Gugercin and A. C. Antoulas, *A survey of model reduction by balanced truncation and some new results*, Internat. J. Control, 77 (2004), pp. 748–766. (cf. p. 3)

[78] G. Guiochon, A. Felinger, D. G. Shirazi, and A. M. Katti, *Fundamentals of Preparative and Nonlinear Chromatography*, Academic Press, 2006. (cf. pp. 4, 12, 20)

[79] M. D. Gunzburger, J. S. Peterson, and J. N. Shadid, *Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1030–1047. (cf. pp. 32, 35)

[80] B. Haasdonk, *Convergence rates of the POD-Greedy method*, M2AN Math. Model. Numer. Anal., 47 (2013), pp. 859–873. (cf. p. 43)

[81] B. Haasdonk, *Reduced basis methods for parametrized PDEs — a tutorial introduction for stationary and instationary problems*, tech. rep., University of Stuttgart, 2014. (cf. pp. 3, 34, 35)

[82] B. Haasdonk, M. Dihlmann, and M. Ohlberger, *A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 423–442. (cf. pp. 6, 41, 43, 78)

[83] B. Haasdonk and M. Ohlberger, *Adaptive basis enrichment for the reduced basis method applied to finite volume schemes*, in Proc. 5th International Symposium on Finite Volumes for Complex Applications, 2008, pp. 471–478. (cf. pp. 6, 41, 78)

[84] ——, *Reduced basis method for explicit finite volume approximations of nonlinear conservation laws*, in Proc. 12th International Conference on Hyperbolic Problems: Theory, Numerics, Application, Providence, USA, 2008, AMS. (cf. p. 33)

[85] ——, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, M2AN Math. Model. Numer. Anal., 42 (2008), pp. 277–302. (cf. pp. 6, 33, 40, 41, 42, 52, 58, 59)

[86] B. Haasdonk and M. Ohlberger, *Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 145–161. (cf. p. 53)

[87] B. Haasdonk, M. Ohlberger, and G. Rozza, *A reduced basis method for evolution schemes with parameter-dependent explicit operators*, Electron. Trans. Numer. Anal., 32 (2008), pp. 145–161. (cf. pp. 38, 48, 53, 88)

[88] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Methuen, 1964. (cf. p. 108)

[89] M. Heinkenschloss, D. C. Sorensen, and K. Sun, *Balanced truncation model reduction for a class of descriptor systems with applications to the Oseen equations*, SIAM J. Sci. Comput., 30 (2008), pp. 1038–1063. (cf. p. 28)

[90] M. W. Hess and P. Benner, *Fast evaluation of time-harmonic Maxwell's equations using the reduced basis method*, IEEE Trans. Microw. Theory Techn., 61 (2013), pp. 2265–2274. (cf. p. 33)

[91] ——, *A reduced basis method for microwave semiconductor devices with geometric variations*, COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, 33 (2014), pp. 1071–1081. (cf. pp. 33, 39)

[92] M. Hinze and S. Volkwein, *Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. Sorensen,

eds., vol. 45 of Lect. Notes Comput. Sci. Eng., Springer-Verlag, Berlin/Heidelberg, Germany, 2005, pp. 261–306. (cf. p. 52)

[93] K.-H. Hoffmann, G. Leugering, and F. Tröltzsch, eds., *Optimal control of partial differential equations. Proceedings of the IFIP WG 7.2 international conference, Chemnitz, Germany, April 20–25, 1998*, vol. 133 of ISNM, International Series of Numerical Mathematics, Birkhäuser, Basel, Switzerland, 1999. (cf. p. 22)

[94] C. Homescu, L. R. Petzold, and R. Serban, *Error estimation for reduced-order models of dynamical systems*, SIAM J. Numer. Anal., 43 (2005), pp. 1693–1714. (cf. p. 52)

[95] R. H. Hoppe and Z. Liu, *Snapshot location by error equilibration in proper orthogonal decomposition for linear and semilinear parabolic partial differential equations*, J. Numer. Math., 22 (2014), pp. 1–32. (cf. p. 67)

[96] H. Hotelling, *Analysis of a complex of statistical variables into principal components.*, J. Educ. Psychol., 24 (1933), pp. 417–441,498–520. (cf. p. 30)

[97] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera, *A successive constraint linear optimization method for lower bounds of parametric coercivity and inf–sup stability constants*, C. R. Math. Acad. Sci. Paris, 345 (2007), pp. 473–478. (cf. pp. 33, 52)

[98] T. Iliescu and Z. Wang, *Are the snapshot difference quotients needed in the proper orthogonal decomposition?*, SIAM J. Sci. Comput., 36 (2014), pp. A1221–A1250. (cf. pp. 43, 52)

[99] A. C. Ionita and A. C. Antoulas, *Data-driven parametrized model reduction in the Loewner framework*, SIAM J. Sci. Comput., 36 (2014), pp. A984–A1007. (cf. p. 26)

[100] K. Ito and S. S. Ravindran, *A reduced-order method for simulation and control of fluid flows*, J. Comput. Phys., 143 (1998), pp. 403–425. (cf. pp. 3, 33)

[101] S. G. Johnson, *The NLopt Nonlinear-Optimization Package*. `http://ab-initio.mit.edu/nlopt`. (cf. pp. 96, 101, 107)

[102] I. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Berlin: Springer, 2002. (cf. p. 30)

[103] S. Kaulmann, M. Ohlberger, and B. Haasdonk, *A new local reduced basis discontinuous Galerkin approach for heterogeneous multiscale problems*, C. R. Math. Acad. Sci. Paris, 349 (2011), pp. 1233–1238. (cf. p. 43)

[104] Y. Kawajiri and L. T. Biegler, *Optimization strategies for simulated moving bed and PowerFeed processes*, AIChE J., 52 (2006), pp. 1343–1350. (cf. pp. 4, 20)

[105] T. Kim, *Frequency-domain Karhunen-Loève method and its application to linear dynamic systems*, AIAA J., 36 (1998), pp. 2117–2123. (cf. p. 34)

[106] D. J. Knezevic, N. C. Nguyen, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for the parametrized unsteady boussinesq equations*, Math. Models Methods Appl. Sci., 21 (2011), pp. 1415–1442. (cf. p. 33)

[107] D. D. Kosambi, *Statistics in function space*, J. Indian Math. Soc., 7 (1943), pp. 76–88. (cf. p. 30)

[108] K. Kunisch and S. Volkwein, *Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition*, J. Optim. Theory Appl., 102 (1999), pp. 345–371. (cf. p. 3)

[109] ——, *Galerkin proper orthogonal decomposition methods for parabolic systems*, Numer. Math., 90 (2001), pp. 117–148. (cf. p. 43)

[110] S. Lefteriu and A. C. Antoulas, *A new approach to modeling multiport systems from frequency-domain data*, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 29 (2010), pp. 14–27. (cf. p. 26)

[111] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, vol. 31, Cambridge University Press, 2002. (cf. pp. 17, 18)

[112] S. Li, L. Feng, P. Benner, and A. Seidel-Morgenstern, *Using surrogate models for efficient optimization of simulated moving bed chromatography*, Comput. Chem. Eng., 67 (2014), pp. 121–132. (cf. pp. 4, 20, 22, 23, 102, 108)

[113] S. Li, Y. Kawajiri, J. Raisch, and A. Seidel-Morgenstern, *Optimization of simulated moving bed chromatography with fractionation and feedback: Part* I. *Fractionation of one outlet*, J. Chromatogr. A, 1217 (2010), pp. 5337–5348. (cf. pp. 4, 20)

[114] S. Li, Y. Yue, L. Feng, P. Benner, and A. Seidel-Morgenstern, *Model reduction for linear simulated moving bed chromatography systems using Krylov-subspace methods*, AIChE J., 60 (2014), pp. 3773–3783. (cf. p. 23)

[115] M. Loève, *Probability Theory*, New York: D. Van Nostrand Company, 1955. (cf. p. 30)

[116] R. Lübke, A. Seidel-Morgenstern, and L. Tobiska, *Numerical method for accelerated calculation of cyclic steady state of Modicon–SMB-processes*, Comput. Chem. Eng., 31 (2007), pp. 258–267. (cf. p. 82)

[117] J. L. Lumley, *The structure of inhomogeneous turbulent flows*, Atmospheric Turbulence and Radio Wave Propagation, (1967), pp. 166–178. (cf. p. 30)

[118] H. V. Ly and H. T. Tran, *Modeling and control of physical processes using proper orthogonal decomposition*, Math. Comput. Model., 33 (2001), pp. 223–236. (cf. p. 35)

[119] Y. Maday and O. Mula, *A generalized empirical interpolation method: Application of reduced basis techniques to data assimilation*, in Analysis and Numerics of Partial Differential Equations, Springer, 2013, pp. 221–235. (cf. p. 49)

[120] Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. H. Pau, *A general multipurpose interpolation procedure: The magic points*, Commun. Pure Appl. Anal., 8 (2009), pp. 383–404. (cf. p. 47)

[121] R. Markovinović and J. D. Jansen, *Accelerating iterative solution methods using reduced-order models as solution predictors*, Internat. J. Numer. Methods Engrg., 68 (2006), pp. 525–541. (cf. p. 83)

[122] M. Mazzotti, G. Storti, and M. Morbidelli, *Optimal operation of simulated moving bed units for nonlinear chromatographic separations*, J. Chromatogr. A, 769 (1997), pp. 3–24. (cf. pp. 98, 103)

[123] V. Mehrmann and T. Stykel, *Balanced truncation model reduction for large-scale systems in descriptor form*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. Sorensen, eds., vol. 45 of Lect. Notes Comput. Sci. Eng., Springer-Verlag, Berlin/Heidelberg, Germany, 2005, pp. 83–115. (cf. p. 28)

[124] M. Minceva, L. S. Pais, and A. E. Rodrigues, *Cyclic steady state of simulated moving bed processes for enantiomers separation*, Chem. Eng. Process.: Process Intensification, 42 (2003), pp. 93–104. (cf. p. 102)

[125] B. C. Moore, *Principal component analysis in linear systems: controllability, observability, and model reduction*, IEEE Trans. Autom. Control, AC-26 (1981), pp. 17–32. (cf. pp. 3, 27)

[126] K. W. Morton and D. F. Mayers, *Numerical Aolution of Partial Differential Equations: An Introduction*, Cambridge University Press, Cambridge, UK, 2005. (cf. p. 58)

[127] J. P. Mota, J. M. Araújo, R. Rodrigues, et al., *Optimal design of simulated moving-bed processes under flow rate uncertainty*, AIChE J., 53 (2007), pp. 2630–2642. (cf. pp. 108, 110)

[128] F. Negri, A. Manzoni, and D. Amsallem, *Efficient model reduction of parametrized systems by matrix discrete empirical interpolation*, J. Comput. Phys., 303 (2015), pp. 431–454. (cf. pp. 39, 48)

[129] N. C. Nguyen, *A posteriori error estimation and basis adaptivity for reduced-basis approximation of nonaffine-parametrized linear elliptic partial differential equations*, J. Comput. Phys., 227 (2007), pp. 983–1006. (cf. p. 52)

[130] N. C. Nguyen, A. T. Patera, and J. Peraire, *A 'best points' interpolation method for efficient approximation of parametrized functions*, Internat. J. Numer. Methods Engrg., 73 (2008), pp. 521–543. (cf. pp. 44, 45, 47)

[131] N. C. Nguyen and J. Peraire, *An efficient reduced-order modeling approach for non-linear parametrized partial differential equations*, Internat. J. Numer. Methods Engrg., 76 (2008), pp. 27–55. (cf. p. 47)

[132] N. C. Nguyen, G. Rozza, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation*, Calcolo, 46 (2009), pp. 157–185. (cf. pp. 6, 33, 34, 42, 52, 58, 71, 72)

[133] A. K. Noor and J. M. Peters, *Reduced basis technique for nonlinear analysis of structures*, AIAA J., 18 (1980), pp. 145–161. (cf. pp. 3, 33, 52)

[134] G. R. North, T. L. Bell, R. F. Cahalan, and F. J. Moeng, *Sampling errors in the estimation of empirical orthogonal functions*, Monthly Weather Review, 110 (1982), pp. 699–706. (cf. p. 30)

[135] A. Odabasioglu, M. Celik, and L. T. Pileggi, *PRIMA: Passive reduced-order interconnect macromodeling algorithm*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 17 (1998), pp. 645–654. (cf. p. 30)

[136] P. Pacciarini and G. Rozza, *Stabilized reduced basis method for parametrized advection–diffusion PDEs*, Comput. Methods Appl. Mech. Engrg., 274 (2014), pp. 1–18. (cf. p. 39)

[137] H. Panzer, T. Wolf, and B. Lohmann, *$\mathcal{H}_2$ and $\mathcal{H}_\infty$ error bounds for model order reduction of second order systems by Krylov subspace methods*, in Proc. European Control Conf. ECC 2013, Zurich, 2013, pp. 4484–4489. (cf. p. 30)

[138] H. K. F. Panzer, *Model Order Reduction by Krylov Subspace Methods with Global Error Bounds and Automatic Choice of Parameters*, PhD thesis, Technische Universität München, München, 2014. (cf. p. 30)

[139] J. Park and I. W. Sandberg, *Universal approximation using radial-basis-function networks*, Neural Comput., 3 (1991), pp. 246–257. (cf. p. 22)

[140] A. T. Patera and G. Rozza, *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations*, MIT, 2007. Cambridge, MA, also available from `http://augustine.mit.edu/methodology/methodology_book.htm`. (cf. pp. 3, 4, 33, 34, 35, 37, 52, 59)

[141] A. Paul-Dubois-Taine and D. Amsallem, *An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models*, Internat. J. Numer. Methods Engrg., 102 (2015), pp. 1262–1292. (cf. pp. 6, 41, 78)

[142] B. Peherstorfer, D. Butnaru, K. Willcox, and H. J. Bungartz, *Localized discrete empirical interpolation method*, SIAM J. Sci. Comput., 36 (2014), pp. A168–A192. (cf. pp. 32, 43, 48)

[143] B. Peherstorfer and K. Willcox, *Online adaptive model reduction for nonlinear systems via low-rank updates*, SIAM J. Sci. Comput., 37 (2015), pp. A2123–A2150. (cf. pp. 32, 48)

[144] J. S. Peterson, *The reduced basis method for incompressible viscous flow calculations*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 777–786. (cf. p. 33)

[145] J. R. Phillips, *Projection-based approaces for model reduction of weakly nonlinear, time-varying systems*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 22 (2003), pp. 171–187. (cf. p. 34)

[146] T. A. Porsching, *Estimation of the error in the reduced basis method solution of nonlinear equations*, Math. Comput., 45 (1985), pp. 487–496. (cf. pp. 3, 33, 52)

[147] C. Prud'homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici, *Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods*, J. Fluids Engrg., 124 (2002), pp. 70–80. (cf. pp. 6, 33, 37, 52)

[148] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, vol. 92 of Unitext Series, Springer, Basel, Switzerland, 2016. (cf. pp. 34, 39)

[149] A. Quarteroni and G. Rozza, eds., *Reduced Order Methods for Modeling and Computational Reduction*, vol. 9 of Modeling, Simulation & Applications Series, Springer, Cham, Switzerland, 2014. (cf. pp. 4, 34, 35)

[150] A. Quarteroni, G. Rozza, and A. Manzoni, *Certified reduced basis approximation for parametrized partial differential equations and applications*, J. Math. Ind., 1 (2011), pp. 1–49. (cf. p. 34)

[151] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, vol. 23, Springer-Verlag Berlin Heidelberg, 2008. (cf. p. 2)

[152] A. Rajendran, G. Paredes, and M. Mazzotti, *Simulated moving bed chromatography for the separation of enantiomers*, J. Chromatogr. A, 1216 (2009), pp. 709–738. (cf. pp. 4, 13, 20)

[153] M. Rathinam and L. R. Petzold, *A new look at proper orthogonal decomposition*, SIAM J. Numer. Anal., 41 (2003), pp. 1893–1925. (cf. pp. 3, 52)

[154] T. Reis and T. Stykel, *Stability analysis and model order reduction of coupled systems*, Math. Comput. Model. Dyn. Syst., 13 (2007), pp. 413–436. (cf. p. 39)

[155] T. Reis and T. Stykel, *Positive real and bounded real balancing for model reduction of descriptor systems*, Internat. J. Control, 83 (2010), pp. 74–88. (cf. p. 28)

[156] M. J. Rewieński and J. White, *A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 22 (2003), pp. 155–170. (cf. p. 34)

[157] M. J. Rewieński and J. White, *Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations*, Linear Algebra Appl., 415 (2006), pp. 426–454. (cf. p. 34)

[158] D. V. Rovas, *Reduced-Basis Output Bound Methods for Parametrized Partial Differential Equations*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003. (cf. pp. 6, 33, 52)

[159] D. V. Rovas, L. Machiels, and Y. Maday, *Reduced-basis output bound methods for parabolic problems*, IMA J. Numer. Anal., 26 (2006), pp. 423–445. (cf. pp. 6, 52)

[160] C. W. Rowley, *Model reduction for fluids, using balanced proper orthogonal decomposition*, Int. J. Bifurcation Chaos, 15 (2005), pp. 997–1013. (cf. p. 34)

[161] G. Rozza, D. B. P. Huynh, and A. Manzoni, *Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants*, Numer. Math., 125 (2013), pp. 115–152. (cf. p. 52)

[162] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Arch. Comput. Methods Eng., 15 (2008), pp. 229–275. (cf. pp. 3, 52)

[163] G. Rozza and K. Veroy, *On the stability of the reduced basis method for Stokes equations in parametrized domains*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1244–1260. (cf. p. 39)

[164] D. Ryckelynck, *A priori hyperreduction method: an adaptive approach*, J. Comput. Phys., 202 (2005), pp. 346–366. (cf. p. 49)

[165] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM Publications, Philadelphia, PA, 2003. (cf. p. 39)

[166] E. W. Sachs and M. Schu, *A priori error estimates for reduced order models in finance*, ESAIM: Math. Model. Numer. Anal., 47 (2013), pp. 449–469. (cf. p. 53)

[167] W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, Springer-Verlag, Berlin, Heidelberg, 2008. (cf. pp. 4, 35)

[168] A. Seidel-Morgenstern, L. C. Kessler, and M. Kaspereit, *New developments in simulated moving bed chromatography*, Chem. Eng. Technol., 31 (2008), pp. 826–837. (cf. pp. 4, 13, 20)

[169] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, *Kriging models for global approximation in simulation-based multidisciplinary design optimization*, AIAA J., 39 (2001), pp. 2233–2241. (cf. p. 22)

[170] J. R. Singler, *New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs*, SIAM J. Numer. Anal., 52 (2014), pp. 852–876. (cf. p. 52)

[171] L. Sirovich, *Turbulence and the dynamics of coherent structures. parts I-III*, Quart. Appl. Math., 45 (1987), pp. 561–590. (cf. p. 3)

[172] J. Strube, S. Haumreisser, H. Schmidt-Traub, M. Schulte, and R. Ditz, *Comparison of batch elution and continuous simulated moving bed chromatography*, Org. Process Res. Dev., 2 (1998), pp. 305–319. (cf. p. 13)

[173] A. Toumi, S. Engell, M. Diehl, H. Bock, and J. Schlöder, *Efficient optimization of simulated moving bed processes*, Chem. Eng. Process.: Process Intensification, 46 (2007), pp. 1067–1084. (cf. pp. 4, 20)

[174] A. Toumi, S. Engell, O. Ludemann-Hombourger, R. Nicoud, and M. Bailly, *Optimization of simulated moving bed and Varicol processes*, J. Chromatogr. A, 1006 (2003), pp. 15–31. (cf. pp. 4, 20)

[175] K. Urban and A. T. Patera, *An improved error bound for reduced basis approximation of linear parabolic problems*, Math. Comp., 83 (2014), pp. 1599–1615. (cf. pp. 52, 59)

[176] K. Veroy and A. T. Patera, *Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori*

error bounds, Internat. J. Numer. Methods Fluids, 47 (2005), pp. 773–788. (cf. pp. 6, 33, 52)

[177] K. Veroy, C. Prud'homme, and A. T. Patera, *Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds*, C. R. Math. Acad. Sci. Paris, 337 (2003). (cf. pp. 6, 33, 52)

[178] K. Veroy, C. Prudhomme, D. V. Rovas, and A. T. Patera, *A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations*, in Proc. of the 16th AIAA computational fluid dynamics conference, vol. 3847, 2003, pp. 23–26. (cf. pp. 33, 40)

[179] S. Volkwein, *Model reduction using proper orthogonal decomposition*, Lecture notes, available from `http://www.uni-graz.at/imawww/volkwein/POD.pdf`, University of Graz, 2011. (cf. p. 3)

[180] K. Willcox, *Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition*, Comput. Fluids, 35 (2006), pp. 208–226. (cf. pp. 3, 49, 59)

[181] K. Willcox and J. Peraire, *Balanced model reduction via the proper orthogonal decomposition*, AIAA J., 40 (2002). (cf. pp. 3, 34, 59)

[182] D. Wirtz, D. C. Sorensen, and B. Haasdonk, *A posteriori error estimation for DEIM reduced nonlinear dynamical systems*, SIAM J. Sci. Comput., 36 (2014), pp. A311–A338. (cf. p. 52)

[183] M. Yano, *A space-time Petrov–Galerkin certified reduced basis method: Application to the Boussinesq equations*, SIAM J. Sci. Comput., 36 (2014), pp. 232–266. (cf. pp. 39, 52, 59)

[184] M. Yano and A. T. Patera, *A space-time variational approach to hydrodynamic stability theory*, Proc. R. Soc. Lond. Ser. A Math., Phys. Eng. Sci., 469 (2013), p. 0036. (cf. pp. 52, 59)

[185] M. Yano, A. T. Patera, and K. Urban, *A space-time hp-interpolation-based certified reduced basis method for Burgers' equation*, Math. Models Methods Appl. Sci., 24 (2014), pp. 1903–1935. (cf. pp. 52, 59)

[186] Y. Yue and K. Meerbergen, *Accelerating optimization of parametric linear systems by model order reduction*, SIAM J. Optim., 23 (2013), pp. 1344–1370. (cf. pp. 22, 23)

[187] Y. Zhang, L. Feng, S. Li, and P. Benner, *Accelerating PDE constrained optimization by the reduced basis method: application to batch chromatography*, Internat. J. Numer. Methods Engrg., 104 (2015), pp. 983–1007. (cf. pp. , 10, 33, 46, 51, 58, 59, 79, 85, 93)

[188] ——, *An efficient output error estimation for model order reduction of parametrized evolution equations*, SIAM J. Sci. Comput., 37 (2015), pp. B910–B936. (cf. pp. , 25, 33, 51, 85)

[189] Y. Zhang, L. Feng, A. Seidel-Morgenstern, and P. Benner, *Accelerating optimization and uncertainty quantification of nonlinear SMB chromatography using reduced-order models*. Submitted. (cf. pp. , 10, 25, 85)